Automated Contract Negotiation and Execution as a System of Constraints

Peyman Faratin and Mark Klein

Center for Coordination Science
Sloan School of Management
MIT
Cambridge MA 02142
{peyman,m_klein}@mit.edu

ABSTRACT

A classification of constraints is proposed that represents not only the individual and negotiated decisions of multiple agents as constraints, but also the exceptions that can occur at execution time. A design of an exception mechanism is then proposed based on task-environment constraints. This mechanism is composed of an informative and a normative component. The informative component functions to update the beliefs of agents about possible exceptions during the negotiation or execution stages of a joint activity. The normative component, on the other hand, places requirements on the agent to reason about such exceptions. The interaction of such an informative mechanism on a bargaining mechanism is elaborated in this paper. A simple additive model of future events is assumed to reasonably model this information. Different agent bargaining strategies, characterized as different attitudes towards the original constraints of the local problem given this belief, are then evaluated for a concession solver in an alternating sequential protocol.

1. INTRODUCTION

The problem of concern is how autonomous computational agents, in a multi-agent system (MAS), can reason and agree on contracts when there are insecurities involved in both the processes of negotiating and executing a contract. The type of negotiation of interest for us is bargaining between a buyer and a seller for a service [6]. Bargaining, in contrast to other cooperative coordination mechanisms involves resolution of conflicting *preferences* between selfish agents where little or no information is shared and there is no objectively correct solution [24, 28]. This bargaining problem belongs to the class of problems where the objective of each agent is to maximize its profit subject to some budgetary limits, a subclass of the general problem of constraint satisfaction. To solve this type of problem both operation research and AI have proposed constraint solvers for a given domain of the constraint (CSP algorithms of AI and Integer Programming techniques of OR for discrete constraints, or continuous optimization techniques for real valued constraints). In this paper the local problem of an agent is assumed to be defined in the pre-negotiation phase where decision variables describing a service and their preferred satisfaction constraints are enumerated. The solution to this local problem is then viewed as an optimization problem constrained by a multi-dimensional system, rather than just a single constraint of a price budget. The negotiation problem is then viewed as a *distributed* constraint optimization problem where a set of variables describing a service are shared among a set of agents [27, 1]. However, because there are dependencies among variable values then the local optimization process needs to simultaneously satisfy both the local and interaction constraints since the joint solution requires the inclusions of other agents' optimization choices. The solution to this distributed optimization problem is what we call a contract. The execution problem in turn is viewed as a commitment problem when agreed to contract (or solutions) in the negotiation phases must be honored. In this paper we motivate the view that commitments are also, like domain and interaction variables, constraints over choices and actions.

The contribution of this work are: i) a taxonomy of constraint types, that can be instantiated at the pre-negotiation, negotiation and execution stages of a joint activity, ii) use of this taxonomy to model an *integrated* contracting system that better describes the mechanisms, the (types and number of) constraints they handle and the stages of the activity they operate, and iii) a design proposal for an exception mechanism for selfish agents that handle task-environment constraints and show the interaction of this mechanism with a bargaining mechanism where agents form their strategies given beliefs derived from such an exception mechanism.

The structure of the paper is as follows. Section 2 briefly motivates the rationales for the approach. This is followed by an in-depth enumeration of three stages of a joint activity in section 3. Constraint types are then identified and classified for each of these stages in section 4. An illustrative example is then given in section 5 to demonstrate concepts from the proceeding sections. A preliminary model of constraint reasoning, given only one informative component of the mechanism, is then presented and evaluated in the penultimate section 6. Finally, the conclusions reached and the future work are enumerated in section 7.

2. THE RATIONALE

Most coordination models only partially solve the overall problem of how agents form *and* execute solutions.[1] Generally, in social systems there is often a chain of *interdependent* decisions and actions (local and/or social) at every stage of a collaborative activity, from preparing to enter negotiation, negotiating, and executing the agreed solution/contract where actions/outcomes in one stage strongly influence/constrain the next. The approach taken here views different collaborative problem solving task as composed of a number of different types of constraints (such as domain, interaction and commitment constraints) each handled by different set of mechanism (decision, coordination and execution mechanisms respectively) occurring at different stages of a collaborative activity (pre-negotiation, negotiation and execution respectively). Under this view different agent architectures, negotiation and commitment protocols can be seen as proposals for mechanisms that handle different types of constraints. However, they often operate *independently* at different stages of a collaborative task. Yet a satisfactory solution must account for all, and not just parts, of this overall coordination problem. In this paper we want to motivate the position that constraints, and mechanisms that model them, can represent a unified modeling framework for the general problem of multi-agent coordination for not only negotiation ([26, 1]), but also execution. Efforts have been made to make the negotiation and the execution phases dependent. For example, in Sandholm's Leveled Commitment Protocol agents reason about decommiting, at either negotiation or execution time, on the agreed outcome at negotiation [20]. Reputation and trust systems also perform this "interconnection" role through providing historical feedback information to earlier stages in collaborative problem solving that constrain choices.

We acknowledge another class of feedback systems, closely related to reputation and trust systems, which we call *exception* systems that function to provide feedback state information to earlier stages of collaborative problem solving. This information system can then be used strategically by agents to reason about all the stages of a collaborative problem solving. This mechanisms makes dependent (through agents' reasoning) not the execution outcome on negotiation outcome, but rather the negotiation outcomes on information about likelihood of possible future states of the world based on previous executions.[2] This information is derived from the observed de-commitments encountered not only during contract execution, but also contract formation stages. We make a further restriction that de-commitments are due to unintentional causes and brought about through a limiting environment called the *task-environment* constraint that occur in negotiation/execution. These stages of a joint activity are described below.

[1]The problem can also be stated as a planning problem where contracts are viewed equivalent to plans [5, 8]

[2]This is motivated by real world contracts where agents do not on priori basis propose "unsignable" contracts given that certain states of the world is likely to hold.

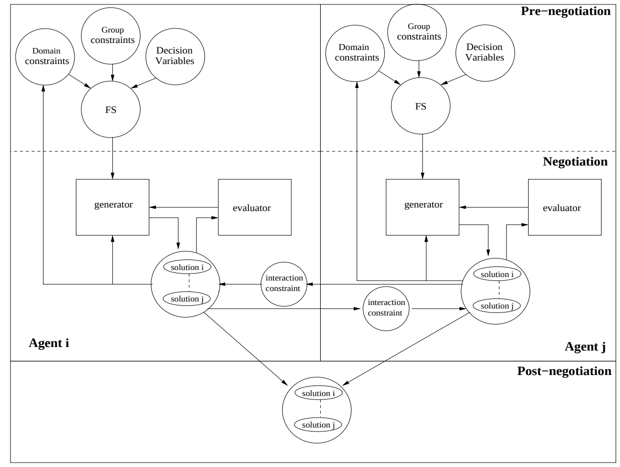

Figure 1: Problem Solving Stages

3. STAGES OF JOINT ACTIVITIES

Any collaborative activities can be usefully classified into the following three stages: i) pre-interaction: a knowledge problem, ii) interaction: gaming problem, iii) post-interaction: commitment problem. These stages are represented for two agents (agent i and j) in figure 1. Boxes, ovals and links represent processes involved in negotiation, data and information flow respectively. The *local* problem of the agent is defined in the pre-interaction phase (or pre-negotiation when the collaborative activity is negotiation). This stage of the collaborative activity can be informally defined as the stage where the agent "gets to know itself and what it wants". More precisely, in a service oriented negotiation this stage of the problem involves agents (buyers and sellers) defining the *feasible set* from a set of decision variables given the domain constraints. The feasible set defines what is possible (or individually rational) before the stage of what is achievable in interaction (or socially rational) [2]. Next, because decisions of agents interact then locally generated solution(s) need to be exchanged between those agents whose choices exhibit interdependency constraints. If the local assignments of values to the decision problem is inconsistent with an n agent assignment requirement then agents need to negotiate to solve this conflict problem. Once the problem of finding a globally consistent assignment has been solved multi-laterally the problem is then transformed to ensuring all agents honour their commitments at execution time.

Sections below expand on each of the above stages.

3.1. THE PRE-NEGOTIATION PROBLEM

The pre-negotiation, or non-interactive, stage of the problem of the agent is a knowledge problem. At this stage the agent attempts to not only define its local problem, but *may* also attempt to solve it independently of interactions with other agents. A local solution is locally consistent assignment of values to a set of decision variable that satisfy some set of constraints. Let $j (j \in \{a, b\})$ represent an agent. Let $I^j = \{i^j_1, \ldots, i^j_n\}$ represent the local n dimensional decision variables, or issues, of agent j. In a service oriented negotiation we restrict ourselves to the problems where both the set and ontology of the decision variables are shared among the agents. That is $I^a = I^b$, in both set members and ontology. Although the assumption of a shared ontology is not necessary for the model it is sufficient to simplify the coordination problem. The assumption that decision variables are shared objects in negotiation is, on the other hand, essential for the type of negotiations we consider which require agents to reason about a shared set of decision variables. This preference based (bargaining) model is contrasted with persuasion, argumentation [24, 11, 15], or information based models and protocols of coordination that do not require a shared decision variable sets. Furthermore, let $C_i^j = \{c_1^j, \ldots, c_l^j\}$ be a set of l dimensional constraints of agent j for each issue i.

The local decision problem of an agent is then defined by the function $f(X^j)$ that *preferentially* satisfies a set of constraints C^j of a set of issues I^j. X^j is interpreted as the most preferred local solution or an assignment of values to each issue ($\forall i^j \in I^j$) that is consistent with the set of constraints for each variable C^j for agent j. This constraint satisfaction problem is similar to the constrained optimization problem of an economic agent. The problem of an economic agent is defined by the following steps. First step involves deriving the feasible (*FS*), or attainable, set from the decision variables. The values of the decision variables must satisfy the constraints placed through the existence of a limited resource. The derived *FS* can be either continuous or discrete. Note, that only technical and non-preferential information is required to determine *FS*. The second step is defining a criteria function (profit, cost, etc.) which suitably reflects the preferences of the agent by associating numbers to elements in the feasible set. Finally, a solution to this problem is sought that optimizes the criteria function (finding the best point in the *FS*) using some mathematical technique (e.g Simplex [3]). Note that to compute X an agent is assumed to know two aspects of the problem: i) a declarative aspect of how to represent its preferences over members of the *FS* and ii) the procedural one of how to solve its local problem (compute f using either some exterior or interior method [13]) given *FS*. That is, classically it is assumed that even before entering negotiations a rational agent is required to know methods for generating the *FS* as well as being able to compute $f()$ (the most preferred) within some metric of boundedness. This assumption has implications on the knowledge complexity of the agent because the proof of completeness and the soundness of the solution to this problems is known to be hard, even in the linear case [19]. Therefore, in the worst case an agent has to solve three hard problems: a local, social and an execution optimization problem.

3.2. THE NEGOTIATION PROBLEM

Once agents have "in mind" a consistent assignment of values to each of their local decision variables, they enter the next stage of a collaborative activity which is negotiation or the modification and checking of consistency of the *joint* set of constraints. The negotiation problem is a gaming problem, where conflicting preferences (or interaction constraints) make the achievement of mutually agreed set of values for a variable difficult to achieve, often in the case of little or no information revelation [6].

In order to achieve conflict resolution any coordination mechanism must specify the following components. A protocol, or rules of interaction, that coordinate the agents at either asocial level (synchronicity of messages for example) and/or social level (protocols that force the selection of a solution that satisfies some criteria) [17]. Given this protocol of interaction the other component of a coordination mechanism is the agents' strategy set. Depending on the given protocol an agent's strategy can be specified as the preferred choices of the individual in how to: i) generate solutions to the local/global problem (the generator in figure 1) and ii) how to evaluate proposals, given the protocol of interaction (the evaluator in figure 1).

3.3. THE EXECUTION PROBLEM

The execution problem is essentially a commitment problem, where mutually agreed to contract must be honored. However, in the real world this is seldom the case. In fact agreements are often violated due to uncertainties. In the real world the number of sources of these uncertainties can be vast and for this reason contracts are said to be *incomplete* [2]. However, we classify uncertainties into:

- arrival of unanticipated ex-post better solutions—here the agent is motivated to decommit its intention to honor the solution and/or

- occurrence of unanticipated failures—here the agent is still bound to the intention to honor the solution, but is incapable to do so, due to for example communication and/or resource failure etc.

Insecurities of the first type are viewed as future events which will result in agents decommiting their solution. Sandholm's leveled-commitment work belongs to this class [20]. It allows agents the flexibility for future negotiation as in the case of no commitments (by dropping their current commitments), but guarantees agents *some* level of security as in the total commitment case. Our interest lies in the insecurities represented by the second class and how agents can be assisted to anticipate, reason about and recover from unforeseen solution execution failures (or exception) during the solution formation period (during negotiation). In the real world there are two possible strategies for handling exceptions. One is reactive where agents act when errors occur at negotiation or execution phases of a collaborative activity. The alternative is pro-active, where agents reason with information (at both the negotiation and execution phases) about the likelihood of failures *at the pre-negotiation and negotiation phases*, as opposed to when the failures actually happen. In section 6.2 we propose an exception mechanism whose normative component is specified by enforcing the agents to adopt the latter exception handling strategy *without* specifying *how* or *what* this strategy is. If negotiation is seen as a constrained search then this normative component modifies the negotiation mechanism by requiring agents to perform search for not only reaching agreements at negotiation time but also for handling exceptions during the pre-negotiation and negotiation phases (for negotiation and execution exceptions respectively). Agents are required to reason over the task-environment constraints, at both the pre-negotiation and negotiation phases, in addition to the already existing domain and interaction constraints respectively. Part of this deliberation can be coordination policies the agents will undertake given the types and probability of constraints occurring. In fact this is line with the common interpretation of a contract—a protocol for actions given certain likelihoods of future events holding.

Once formed, either objectively (centrally) or subjectively (locally), the agents must then reason over not only over the domain problem (the content of the contract) but also the interaction constraints as well as the exception related information. Reasoning is viewed as a constraint satisfaction problem. Section below details the set of decisions involved as different system of constraints.

4. TYPES AND SOURCES OF CONSTRAINTS

Decision making and execution at all three phases of the problem solving are limited by one or more different types of constraints that have different origins and properties. We classify constraints into: i) *domain* constraints—occurring during decision making, ii) *uncontrollable task-environment* constraints—occurring during decision enacting, iii) *controllable task-environment* constraints—occurring during decision enacting, iv) *interaction* constraints—occurring during decision making and v) *group* constraints—occurring during *both* decision making and enacting

Each constraint types are elaborated on below.

4.1. DOMAIN CONSTRAINTS

Domain constraints are *local/endogenous* restrictions on the local decision making, defined at the pre-interaction stage of a collaborative problem solving. These constraints are constraints not over actions but rather decisions. For example, the first problem of an aeroplane engine engineer is to choose a set of values for a system of variables that describe the characteristics of the engine, such as its weight, its height. A minimal unary constraint is the domain, or the reservation value, of each negotiation issue, represented as $rv_i^j \in [min_i^j, max_i^j]$. Another unary constraint may be the relative importance, or the weight, of an issue relative to the other issues, denoted as $w_i^j \in [0,1]$, the weight of issue i for agent j restricted to values within the interval of $0, 1$. Another possible constraint are the binary/nary dependencies between the decision variables.

Uncontrollable task-environment, controllable task-environment, interaction and group constraints, on the other hand, are constraints that are exogenously defined.

4.2. TASK-ENVIRONMENT CONSTRAINTS

The information provided by the exception mechanism is based on a model of what we call *task-environment* constraint. These constraints are the set of *environment* constraints encountered by an agent during some task execution. The task may be the process of negotiation itself during the negotiation phase, or else the execution of the agreement at the execution phase of the collaborative activity. We distinguish between uncontrollable and controllable task-environment constraints to make explicit the difference between environmental constraints where the locus of control of the events is not the agent and constraints where locus of control of the events is one, other or both of the dyad or some other party. This distinction, specifically the latter, is important when modeling roles, responsibilities and commitments in a MAS.

Uncontrollable task-environment constraints is the set of environmental constraints that are present whenever the agent acts in the world, *independently of the presence / absence of other agents*. Uncontrollable task-environment constraints are natural restrictions on *actions* and are manifested during either the interaction (communication actions) or the post-interaction phase (execution actions). For example, some random external events (represented as a random shock that is beyond the control of the agent or any other agent, e.g. an earthquake) may interfere with and impede with the actions involved in the process of agreeing on the plane design (negotiation) or the making of the engine when the artifact engineer comes to actually make the engine (execution). Conversely, controllable task-environment constraints is the set of constraints that are present whenever the agent acts in the world *in the presence of other agents* and, like uncontrollable task-environment constraint can occur at either the negotiation or execution phases. For example, communication links might fail between an engine and a wing engineer, or alternatively their messages might be misrepresented, or they may lack a shared ontology. In addition to these communication constraints occurring at the negotiation phase the jointly agreed solution when executed at the post-negotiation phase can be subject to further environmental constraints. Resources necessary to make the engine may be late in arriving, or are of the wrong specifications (or flow exceptions). Shared resources (e.g. construction tools, cranes, etc.) may be wrongly allocated to / being over used by other agents (share exception), etc. The important point to note is that the agent(s) themselves are the locus of control of this type of constraint and not some random shock that can not be controlled. Central to this observation is the notion that prevention of exceptions is the responsibility of the agent performing a role and is modeled as commitments where agents are committed to ensuring that exception does not occur [21, 10, 23]. Control of such unwanted events is achieved in a distributed way by either agents taking appropriate actions in response to such events or else ensuring to honour their agreed commitment at execution phase. In either case, the exception mechanism provides contingency information over classes of likely exceptions. Agents are then normatively required to reason about avoidance, detection and resolution protocols. In such a framework, the normative component of the exception mechanism requires agents to negotiate over the adoption of task-environment constraints as part of their agreements.

4.3. INTERACTION CONSTRAINTS

Conversely, interaction constraints are *preference* or requirement constraints that are present whenever an agent interact with other agents. We dissociate interaction constraints from (uncontrollable / controllable) task-environmental constraints to distinguish between constraints that are due to (un)controllable exogenous events (uncontrollable / controllable task-environmental constraints) versus constraints on local choice whose origin is the *preference* of other agents (interaction constraints). Indeed, this is consistent with the view that negotiation is a social mechanism that achieves constraint satisfaction in a distributed manner. Additional mechanisms (such as reputation or trust systems) can be provided that regulates the initial conflict level of the negotiation mechanism. An implicit assumption here is that interaction constraints are *always* present, hence why a coordination mechanism is almost always required whenever there are interactions (with the exception of when conflicts are removed at compile time, e.g social laws, [22]). The variables of the mechanism are then the type of coordination (negotiation, persuasion, argumentation, social laws) and the level of the conflict (moderated by other mechanisms). A contribution of this work is to address the need for explicit reasoning over task-environment constraint information in a similar way a negotiation mechanism attempts to solve a distribute interaction constraint problem. The approach adopted is to modify a negotiation mechanism by *including* or *combining* decision making over task-environment constraints with the decision making over the interaction constraints.

4.4. GROUP CONSTRAINTS

Finally, in a social system the local actions/decisions of an agent can be constrained not only by other individuals, or likelihood of events/outcomes being true, but also by some collectively negotiated/imposed norm. Such norms can be used to normatively specify: i) the acceptable members of the *FS* (shown in figure 1 as the directed edge connecting the group constraints to the *FS*), and/or ii) the processes of the negotiation and execution. Government rules and regulations and organizational standards and practices are examples of such group constraints. Note, that whereas in a social system the “adoption” of social and group constraints is (more or less) an individual’s choice, the same cannot generally be said for individual task-environment or domain constraints since an agent cannot negotiate with “mother nature” over the terms and the environment of its local or problem execution. The implication of this observation is that depending on the environment of an agent the solution set for problems that account for both domain and task-environment constraints are likely to be smaller in size than solutions that ignore the latter type of constraint.

Overall, social systems exhibit a hierarchy of constraints from different sources, ranging from the domain to extraneously occurring or *imposed* constraints on local behaviour/decisions. We will call agents that reason with either the domain, task-environment, interaction or group constraints as asocial, responsive, social and societal agents respectively. Thus, an asocial engine designer may have “in mind” an ideal design that satisfies *only* the local domain constraints of its problem. Conflicts then, in this context, is when the current decision variable solutions are inconsistent with other agents’ current choices. On the other hand, a social engine designer may have “in mind” an ideal design that satisfies *both* the domain constraints *and* the constraints of other agents encountered in previous design cycles. In this context conflicts arise when the *current* decision variable solutions are inconsistent with other agents’ *current*, and not previous, choices. One hypothesis is that social agents, compared to asocial agents, are expected to spend less time coordinating their choices when the frequency of change in the dynamics of the system is low.

5. CONSTRAINTS & REASONING: AN ILLUSTRATIVE EXAMPLE

As an illustrative example of the concepts above consider two agents a and b, negotiating over two issues x and y. An offer is represented by a pair of x, y values as (x, y). Assume that the local problem of agent a is defined by the minimization of a simple linear objective function $x - y$, that satisfies a conjunction of linear arithmetic constraints. More precisely (in normal form):

$$\begin{aligned} \text{minimize } & x - y \\ \text{such that } 1 \leq & X \leq 3, \\ 0 \leq & Y, \\ 2Y - X \leq & 3 \end{aligned} \tag{1}$$

Conversely, assume for simplicity that the agent b has the same set of constraints but its local problem is to *maximize*, rather than minimize, the objective function $x - y$. The feasible set of outcomes, that satisfy each agent’s constraints, and the optimal solutions are shown in figure 2. The dash lines across the polygon of the feasible set represent the contours of the objective function for different values. The arrows represent the direction that maximizes the objective function.

Assume the rules of the interactions is based on Rubinstein’s alternating sequential protocol [18], where agent a offers a locally generated deal at time t^1. At the next time step t^2 agent b can either accept or reject the offer of a. If b rejects then it can counter-propose another locally generated offer. Assume, given this protocol, that a’s only rational strategy in the presence of no other information is to offer a first deal that maximizes its objective function *based only on its domain constraints* (given by equation in 1) at time $t = 1$. This is shown as the unique value X^1. Call this evaluation of a concrete offer, X^t, at time t by an agent j

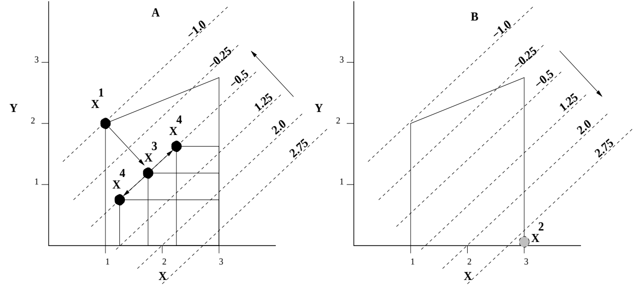

Figure 2: The Feasible Set and Optimal Outcomes for Agent a (left) and Agent b (right).

the aspiration level Θ^t_j. Further assume that b also offers a deal that maximizes its objective function X^2 based only on its domain constraints. Agent a now has a choice based on the information just received, and reasons about what to offer next. Whereas agent $a's$ last offer (X^1) satisfied only its domain constraints now a has to reason about not only its local domain constraints, but also the new constraint that $b's$ offer is not acceptable to a given the current constraint set represented by its aspiration level Θ^t_a (given by 1). That is, a must include this new interaction constraint into its deliberations if a deal is to be reached. The choices of the a essentially amounts to solving the same problem as given in equation 1 but with different (interactions) constraint sets. Figure 2 shows two different strategies [6]. One strategy is to lower its aspiration level and offer X^3 (a concession since the value of the objective function is now lower).[3] Assume that it is also rejected by b. The other strategy of a can be to again solve the same optimization problem but with a different constraint set *that does not effect the level of the objective function* (maintain the same aspiration level). X^4 are two such solutions from the set of all possible solutions that lie on the same optimization contour as X^3 (see [6] for details of the algorithm for this type of reasoning). A solution to the interactive problem is reached when the contours of each party are in contact with one another. In general, different aspiration levels represent different demands on the satisfaction levels of objective function (or points along what is called the iso-curves [2]).

Deliberation over task-environment and/or group/social norms constraints is, in a similar manner to the above, equivalent to solving the optimization problem but with different system of constraints. For example, group norms may normatively specify that offers over x can not exceed 2 ($x \leq 2$). Alternatively, $a's$ previous interactions with b has resulted in deals distributed near pairs of values $(2, 1)$. Alternatively, this knowledge can also be provided by a trusted third party in cases of lack of any previous interactions. Whereas group norms *normatively* delimit portions of the FS, an agent is *not* restricted in how it deliberates over the information about likely exceptions occurring at execution time. Furthermore, if negotiation is viewed as a search process over the space of the possible outcomes specified by the FS then constraining parts of this set at the pre-negotiation time is equivalent to consistency checking, where the delimited areas are not visited during negotiation given the likelihood of an exception. However, this choice of the FS size is a strategy of the agents and is contingent on the exception probabilities. *How* agents define their local constraint is their choice.

In summary, deliberation over domain, interaction, norms and exceptions is modeled as a system of constraints. Reasoning is then a repeated optimization of a progressively smaller FS.[4] Varied solution sets are specified by specifying different aspiration levels of an objective function which in turn represent satisfaction of different set of constraints. The specification of different levels of objective function is interpreted as negotiation reasoning and is equivalent to goal programming techniques [12].

Finally, the importance of the role of an interaction protocol in the reasoning is highlighted when the alternating sequential protocol is compared with others. For example, interaction constraints are not explicitly part of the optimization problem when agents make deals using a one-shot simultaneous protocol. Agents can reason about others choices (the problem of infinite regress) but incentive mechanisms such as Vickrey auction protocols attempt to induce agents to reveal their true reservations thereby obviating the need to perform search.

6. THE INSECURITY PROBLEM

In this section we want to investigate how the informative part of the exception mechanism can be integrated in to the local agent reasoning during the pre-negotiation phase of the joint problem solving. The normative component of the mechanism during the pre and negotiation phases are deferred to future work. Specifically, in this section we assume a simple additive model of exceptions and concentrate on developing and testing different agent strategies.

6.1. A BARGAINING MODEL

The exception mechanism is described for a concrete negotiation mechanism. The details of this negotiation mechanism is as follows. Negotiation is restricted to bi-lateral interactions conducted between two agents a and b, over a set of issues I. Agents exchange a single offer at each time step of negotiation. Offers over all issues are denoted by X. X minimally has to satisfy the domain of all issues. We call this the reservation value, where $\forall x^j_i \in [min^j_i, max^j_i]$ and $x^j_i \in I\!R$. Agents are assumed to offer contracts that respect their true constraints hence incentive mechanisms is assumed unnecessary. Discrete domain constraints together with other non-domain constraints such as time limits are reported in [14, 7, 6]. Offers are evaluated locally using a utility function that captures how well an offer X satisfies

[3]In fact at $t = 2$ agent a can either "hold firm" or concede since X^1 is the single solution to the problem.

[4]Note, the solution of the original problem may be based on exterior algorithms such as the Simplex [3]. However, subsequent optimization, although also possibly based on exterior method, can be interpreted as interior methods [25] since the solution sought is inside the original constraint set.

the reservation domain constraint. A simplifying assumption made is that utility of satisfaction of a contract is the linear combination of the satisfaction of the constraints of each issue. Conflicts are when the satisfaction of one agent's constraints is negatively correlated with the satisfaction of the other agent's constraints.[5] Offers are generated locally and strategically via different solver/ mechanisms. *Negotiation* deliberation involves generation mechanisms that operate *given* both a hard domain constraint (such as the reservation value) and interaction constraints. Mechanisms can be based on the successive modification of the acceptance level that are relaxed (a soft constraint) or alternatively simultaneously demanding and relaxing *individual* acceptance levels of decision variables that produce the same overall acceptance level for an offer. For exploratory purposes we assume the first—a concession generation mechanism where the rate of satisfaction of all domain constraints over successive offers is constantly reduced/relaxed at some rate (see [6] for choices of other generation mechanisms). This is represented as lower utility values for successive offers. Finally, the *negotiation* deliberation execution cycle is assumed to be regulated by the normative rules of an alternating sequential protocol (described above in section 5) where single offers are generated and evaluated locally and communicated to the other party.

6.2. EXCEPTION KNOWLEDGE

The general approach is to design an exception mechanism, implemented and executed by some trusted third party, that transforms the decision problem of an agent from uncertain decisions into risky decision choices. Uncertainty and risks are informally defined as the lack of any information to condition decisions on and some distribution of information, typically a probability distribution, for basis of decision making respectively. This informative component of the mechanism provides not only the types (or classes) of possible contract exceptions that can occur at execution time, but also their associated probabilities. In addition to an informative component the exception mechanism includes a normative component that enforces a protocol of exception reasoning based on three additional exception handling stages (exception anticipation, avoidance, detection and resolution [10, 4]).

We propose that deliberation over *exception* can be represented as modification of the domain constraints (here the reservation values) not during the negotiation, but rather during the pre-negotiation phase. The effect of this deliberation is to selectively modify, conditioned on the probability of exceptions, the original FS for an issue (derived from the original set of reservation constraints and referred to as C_i^j) before the negotiation phase. For example, if the problem of a buying agent is described as the maximization of profit subject to some *initial* monetary constraints, then given the knowledge that a failure of a deal is possible then an agent may be willing to pay less of this initial endowment for the risky event and another may be prepared to pay more. In either case the effect of this knowledge can be modeled by dynamic modification of the domain constraints (the reservation values) which in turn affect the shape and size of the FS. The preference of which strategy to execute is said to be the attitude of the agent given the available information. We model this attitude by dynamic modification of the domain reservation constraint for each issue i for an agent j by the following decision rule:

$$\begin{array}{ll} [min_i^j, (max_i^j - (max_i^j - min_i^j)\alpha_i^j)] & \text{If } U_i^j \text{ is decreasing} \\ [(min_i^j + (max_i^j - min_i^j)\alpha_i^j), max_i^j] & \text{If } U_i^j \text{ is increasing} \end{array}$$

where $\alpha_i^j \in [0,1]$ in the above constraint modification rule is defined as:

$$\alpha_i^j = min(1, \left(\frac{(max(P - P_{min}), 0)}{P_{max} - P_{min}}\right)^{\frac{1}{\beta_i^j}}) \qquad (2)$$

where P, P_{min} and P_{max} represent the current, minimum and maximum believed probabilities of failure respectively. Normally P_{min} and P_{max} are restricted to values 0 and 1 respectively. The attitude of agent a towards this belief is modeled by $\beta_i^j \in \mathbb{R}^+$ for all issues j. When $\beta_i^j < 1$, α_i^j approaches 1 fastest when P tends towards 1. Conversely, when $\beta_i^j > 1$, α_i^j approaches 1 fastest when P is low and remains at an asymptote for increasing values of P. Finally, when $\beta_i^j = 1$ α_i^j approaches 1 linearly with increasing values for P. The result of the decision rule is to shrink the FS by an amount proportional to α_i^j that itself is regulated by β_i^j. Note that the lower the value of β_i^j the closer to modified reservation domain to the original reservation C_i^j for most values of P. Since a utility function ranks the FS as a function of the reservation values then increasing values for β_i^j means the limits of the utility function are defined over a smaller domain because the decision rule reduces the FS more with increasing values for β_i^j. Hence the chances of other agents' offers being "within" the domain of issue j is reduced with increasing values of β_i^j (because FS is smaller). Therefore, lower values for β_i^j encodes the attitude of agent j who is willing to "spend more of its initial budget" on issue i for a deal, even though there are higher risks. This risk attitude for an issue is likely to be adopted if it is compensated for by higher returns for another issue $k \neq i$. That is, agents define the trade-off limits of their local optimization problem given the probability of failure during the pre-negotiation.

Four points require mentioning. Firstly, for simplicity P is assumed to be defined by a linear combination of the probability of all exception classes:

$$P = \sum_e^n w_e p_e$$

where w_e and p_e are the importance of the individual exception case e and the probability of its occurrence respectively. Secondly, nothing is said about who computes P. One possible model of P can be based on objective empirical observations by some trusted third party [4]. Alternatively, agents

[5] This is referred to as zero-sum games in the theory of games [2].

can form their own subjective estimations of P. Thirdly, the costs and benefits (on both local and social benefits) and the incentive mechanisms for local adoption of an objectively formed P is not addressed here. We want to investigate a model of failure deliberation given some probability distribution. Finally, to handle the combinatorial problem of enumerating all possible exceptions, we restrict ourselves to a finite set of exceptions that are domain independent and agent oriented negotiation or execution exceptions [4].

6.3. EXPERIMENTS AND RESULTS

The above model of exception attitude was empirically evaluated to determine the effect of exception knowledge on the bargaining behaviour of agents. The

Results of exception deliberation are presented in figures 3, 4 and 5 for symmetric interaction between two agents with values of $\beta = 5, \beta = 1.0$ and $\beta = 0.5$ respectively for all issues. The assymetric attitudes is shown in figure 6. Final agreed contract are represented by the solid circle for values of P incremented in steps of 0.2 from 0.0 to 0.6. The solid line connecting the two axis represents the pareto-optimal line [2]. Pareto optimality is defined in the following manner. Suppose there are two outcomes $\mathbf{x}$ and $\mathbf{y}$ such that they both belong to the feasible set, $\mathbf{x}, \mathbf{y} \in FS$. If $U_i^j(\mathbf{y}) \geq U_i^j(\mathbf{x})$, for both a and b, but $\mathbf{y}$ is strictly preferred for at least one agent, $U_i^j(\mathbf{y}) > U_i^j(\mathbf{x})$ for $j \in \{a, b\}$, then the outcome $\mathbf{x}$ is not pareto optimal. This is formally represented as a function that given the game defined by the pair FS and $\mathbf{x_c}$ does not select $\mathbf{x}$—i.e., $f(FS, \mathbf{x_c}) \neq \mathbf{x}$. Pareto-optimality is a measure of the limits of efficiency that can be reached in negotiation. Efficiency is defined as the maximization of the addition of individual utilities. The region below the efficient frontier is the set of contracts X exchanged between the agents.

The symmetric results are discussed first. The important point to note is that the *actual* outcomes are a function of the solver/generation mechanism, the evaluator and the protocol of interaction. Therefore, different execution traces are possible if we implemented a different solver. For instance, the concession solver/mechanism cannot find a contract when both a and b have attitudes defined by $\beta_i^a = 1$ and $\beta_i^b = 1$ to $P = 0.6$ (figure 4). However, another solver may find a contract. The concession solver dependent results show that the lower the value of the pair (β^a, β^b), the higher the likelihood of an agreement with higher joint utilities. In other words, more contracts are reachable and the social welfare is increased more if the product of the C_i^a and C_i^b changes less with increasing P. Conversely, fewer contracts are reachable if the reverse is true. However, the interesting point about the data set is instead the input into the solver, defined by the region bounded by the pareto-optimal line. The results show that over-constraining the solver changes the curvilinear shape of the pareto-optimal line to a straight line. The implication of this result is that over constraining the FS results in more competitive (or distributed) negotiations as opposed to "win-win" (or integrative) negotiations. In the latter case each of the distributed solvers can simultaneous, and possibly mutually, satisfy their local constraints. However, this is not possible in the former case [6]. The argument is as follows. When the pareto-optimal line is described by a straight line then the agents' payoffs are perfectly negatively correlated. Then, a contract that increases the utility of one agent decreases the utility of the other. This is referred to as distributive bargaining [16]. Here *all* the possible outcomes lie on or below the pareto-optimal line. Furthermore, assuming linear conflicting utility functions for the negotiation participants, the sum of each outcome is 1 (i.e., it is a *zero-sum game* [9]).

On the other hand, when the pareto-optimal line is described by a curvilinear line then the sum of the individual utilities for a contract does not necessarily add up to 1. That is, in this *non-zero-sum game* it is possible to find contracts in which some of the constraints on some issues are satisfied more (higher utilities) and others are satisfied less (lower utility). Furthermore, this increase may benefit one or both of the negotiation participants simultaneously. Now the only points on this line where the sums of the individual values add to 1 is at the point of connection with the x and y axis. Different points along the line then do not necessarily sum to 1 and do not necessarily have the same addition. This contrasts with the distributive bargaining case where there is no scope for improving one score without decreasing the score of the negotiation opponent. Distributive bargaining is generally a feature of competitative interactions when only a single constraint is being optimized (such as the price of a good). In integrative bargaining, on the other hand, better social outcomes are achievable when multiple constraints are being optimized simultaneously by all solvers. Therefore these results suggest that the strategy of over constraining each of the multi-dimensional constraints results in optimization problem of a single constraint whose satisfaction is perfectly negatively correlated with the other's optimization problem.

The observed results for the assymetric case are presented in figure 6 for $(\beta^a = 0.5, \beta^b = 1)$. Assymetric cases are interesting since the assumption that agents have the same strategy *and* the same belief is strong where agents are unlikely to adopt the same strategies in interactions. In these studies we keep the latter assumption but relax the former. Results show that negotiations between more "cautious" (increasing β) with a less "cautious" agent results in reaching more contracts for higher exception probabilities. For example, at $P = 0.4$ the space of possible solutions is much larger when an agent with $\beta = 1$ negotiates with another with a $\beta = 0.5$ (figure 6) than when the same agent negotiates interacts with a symmetric type (figure 4). These results imply that the possibility of contracts, represented by the FS, is a function of the interactive attitudes of the agents. Here, for example, the results show that interactions with less cautious agents results in finding contracts that the agent would not have committed to if in interactions with another type. Therefore, assymetries appear to affect the dynamics of contracting.

In summary, the results suggest that if high values of β

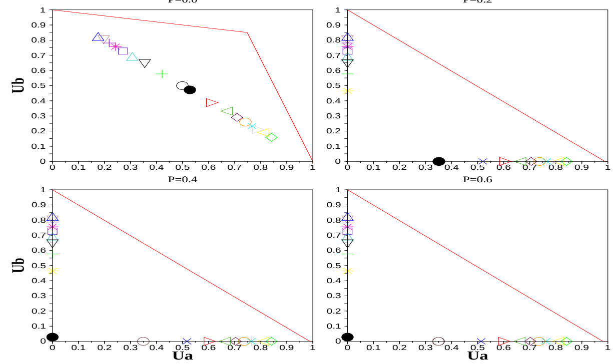

Figure 3: Process of Negotiation with $\beta^a = 5$ and $\beta^b = 5$ for all i

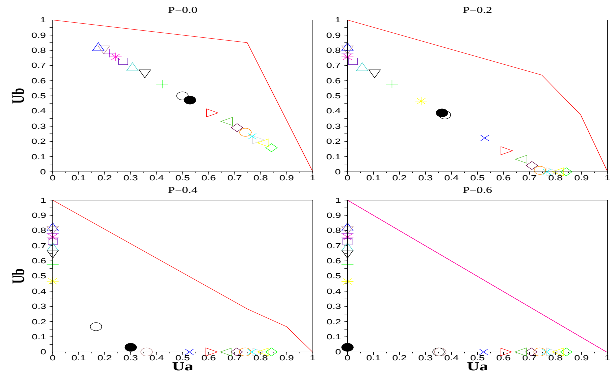

Figure 4: Process of Negotiation with $\beta^a = 1$ and $\beta^b = 1$ for all i

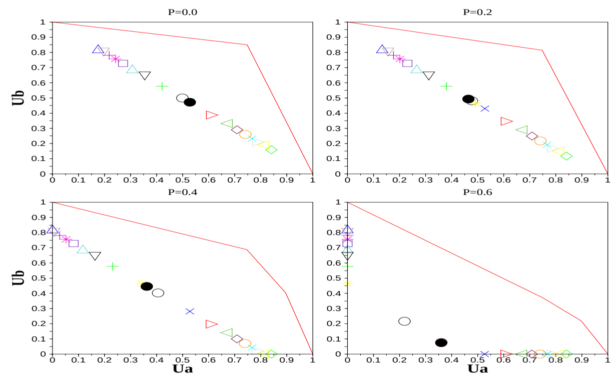

Figure 5: Process of Negotiation with $\beta^a = 0.5$ and $\beta^b = 0.5$ for all i

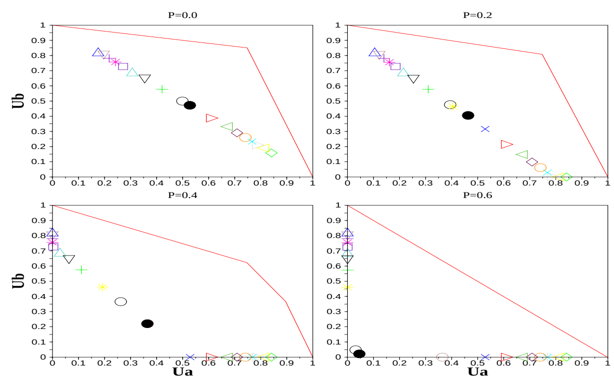

Figure 6: Process of Negotiation with $\beta^a = 0.5$ and $\beta^b = 1$ for all i

are to be interpreted as being "prudent" (or cautious), then the outcomes reached are not good, *if the metric of goodness is maximization of a social welfare function*. However, another metric of goodness maybe needed to align the observed results with the common intuition that to be prudent when there are risks is good. This is a subject of future research where a goodness function is sought that quantifies the contributions of cautiousness of agents in signing contracts given the likelihood of failure. In such context overconstraining the solver may increase the social welfare function.

7. CONCLUSIONS AND FUTURE WORK

We have presented a taxonomy of how deliberation at different stages of a joint problem solving can be represented as a distributed satisfaction of a system of constraints. A design and a rationale for an exception mechanism was also presented. This mechanism is composed of an informative component, used to update the local beliefs of agents, and a normative component that enforces reasoning over exception during negotiation. A simple negotiation problem, viewed as the linear optimization of a system of issue constrains (represented as integer or real valued arithmetic constraints), was then used to evaluate different agent strategies differentially conditioned on the informative component of the exception mechanism. It was shown that information provided by the exception mechanism can be strategically used to prune subjectively believed unreachable contracts. We found that the size of the social welfare possibilities, measured as the sum of individual utilities, increases in interactions with strategies that did not over constrain their local optimization problem with increasing exception probabilities.

The main focus of future work is further developing the informative and normative components of the exception mechanisms. In particular, more sophisticated models of exception events is required to support more flexible reasoning over contract failures. Furthermore, the assumption made that agents willingly adopted the exception information is not strictly true in real social systems. The problem of how the group comes to adopt the objectively derived information also needs to be addressed. Different beliefs are hypothesized to lead to different outcomes. One possible solution to this problem is to append the informative component of the exception mechanism with an escrow (or insurance) component that enforces penalties on the agents when contracts fail. This incentive mechanism then indirectly motivates the agents to adopt beliefs from the objectively estimated probabilities. In addition to this the normative component of the mechanism needs to be further developed. The normative component is seen as including a number of additional negotiation and execution constraints in to the set of negotiation issues. These issues, describing the exception detection, avoidance and resolution mechanisms, are then used as additional constraints for agents to negotiate exception detection, avoidance and resolution protocols that minimizes the loss of the agreed contracts.

REFERENCES

[1] M. Barbuceanu and W. Lo. A multi-attribute utility theoretic negotiation architecture for electronic commerce. In *Proceedings of the Fourth International Conference on Autonomous Agents, Barcelona, Spain (Agents-2000)*, pages 239–247, Barcelona, Spain, 2000.

[2] K. Binmore. *Fun and Games: A Text on Game Theory*. D.C. Heath and Company., Lexington, Massachusetts, 1992.

[3] G. Dantzig. Maximization of a linear function of variables subject to linear inequalities. In T. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 359–373. John Wiley & Sons, 1951.

[4] C. Dellarocas and M. Klein. An experimental evaluation of domain-independent fault handling services in open multi-agent systems. In *Proceedings of International Conference on Multi-Agent Systems (ICMAS-2000), July 2000, Boston, MA*, 2000.

[5] E. Ephrati and J. S. Rosenschein. Multi-agent planning as search for a consensus that maximises social welfare. In C. Castelfranchi and E. Werner, editors, *Artificial Social Systems — Selected Papers from the Fourth European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW-92 (LNAI Volume 830)*, pages 207–226. Springer-Verlag: Heidelberg, Germany, 1994.

[6] P. Faratin, C. Sierra, and N. R. Jennings. Using similarity criteria to make negotiation trade-offs. In *Proceedings of the International Conference on Multiagent Systems (ICMAS-2000), Boston, MA.*, pages 119–126, 2000.

[7] P Faratin, C Sierra, and N.R Jennings. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3–4):159–182, 1998.

[8] M. Georgeff. Communication and interaction in multi-agent planning. In *Proceedings of the National Conference on Artificial Intelligence*, pages 125–129, Washington, D.C., 1983.

[9] R. Gibbons. *A Primer in Game Theory*. Harvester Wheatsheaf, New York, 1992.

[10] M. Klein. A Knowledge-Based Approach to Handling Exceptions in Workflow Systems. *Journal of Computer-Supported Collaborative Work. Special Issue on Adaptive Workflow Systems.*, 9(3/4), 2000.

[11] S. Kraus, N. Nirkhe, and K. Sycara. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence Journal*, 104(1–2):1–69, 1998.

[12] S.M. Lee. *Goal Programming for Decision Analysis*. Auerbach Publishers, Philadelphia, 1972.

[13] K. Marritt and P.J. Stuckey. *Programming with Constraints: An Introduction*. MIT Press, Cambridge, Massachusetts, 1998.

[14] N. Matos, C. Sierra, and N. R. Jennings. Determining successful negotiation strategies: an evolutionary approach. In *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS-98), Paris, France*, pages 182–189, 1998.

[15] S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

[16] H. Raiffa. *The Art and Science of Negotiation*. Harvard University Press, Cambridge, USA, 1982.

[17] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter*. The MIT Press, Cambridge, USA, 1994.

[18] A. Rubinstein. Toward a focal point theory of bargaining. In A.E. Roth, editor, *Game-Theoretic Models of Bargaining*, pages 99–114. Cambridge University Press, NY, 1985.

[19] S. Russell and E. Wefald. *Do the Right Thing*. The MIT Press, 1991.

[20] T.W. Sandholm. Distributed rational decision making. In G. Weiss, editor, *Multiagent Systems*, pages 201–259. The MIT Press, Cambridge, Massachusetts, 1999.

[21] S. Sen. The role of commitment in cooperative negotiation. *International Journal on Intelligent Cooperative Information Systems*, 3(1):67–81, 1994.

[22] Y. Shoham and M. Tenneholtz. On the synthesis of useful social laws for artificial agent societies (preliminary report). In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 276–281, San Jose, 1992.

[23] M. P. Singh. An Ontology for Commitments in Multiagent Systems: Toward a Unification of Normative Concepts. *Artificial Intelligence and Law*, 7:97–113, 1999.

[24] D. N. Walton and E. C. Krabbe. *Commitments in Dialogue: basic Concepts of Interpersonal Reasoning.* State University of New York Press, New York., 1995.

[25] S. Wright. *Primal-Dual Interior-Point Methods.* SIAM Press, 1996.

[26] M. Yokoo, E.H. Durfee, T. Ishida, and K. Kuwabara. The distributed constraint satisfaction problem: Formalization and algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 10(5):673–685, 1998.

[27] M. Yokoo and T. Ishida. Search algorithms for agents. In G. Weiss, editor, *Multiagent Systems*, pages 165–201. The MIT Press, Cambridge, Massachusetts, 1999.

[28] O. R. Young. *Bargaining: Formal Theories of Negotiation*. University of Illinois Press, Urbana, 1975.