Towards a Generic Negotiation Model for Intentional Agents

Fernando Lopes*, Nuno Mamede**, A. Q. Novais*, Helder Coelho***
*INETI, Est. Paco Lumiar, 1699 Lisboa Codex, Portugal
{flopes, anovais@dms.ineti.pt}
**IST, Av. Rovisco Pais, 1049-001 Lisboa, Portugal
Nuno.Mamede@acm.org
***Fac. Ciências, Campo Grande, 1700 Lisboa, Portugal
hcoelho@di.fc.ul.pt

Abstract

Intentional agents are charged with generating and executing intentionally plans of action towards the achievement of their goals. They operate in multi-agent systems and situations often arise in which their plans conflict with the plans of other agents. The predominant process for resolving conflicts is negotiation. This paper presents a generic negotiation mechanism, a new approach for defining a structure for negotiation problems, a method for generating negotiation proposals, and methods for generating counterproposals either by making or not making concessions.

1. Introduction

Intentional agents are autonomous computational processes with internal structures (mental states) composed of beliefs (facts about the environment and the other agents), goals (world states to be achieved), plan templates (procedures for achieving specific world states), and intentions (commitments to achieve world states or to perform actions). The agents are belief-desire-intention agents. The beliefs and the plan templates represent their informational states. The goals and intentions capture, respectively, their motivational and deliberative states.

The agents generate and execute intentionally plans of action. They operate in multi agent systems and situations often arise in which their plans conflict with the plans of other agents. Conflict resolution is crucial for achieving multi-agent coordination. The predominant process for solving conflicts is negotiation - the process by which two or more parties verbalize contradictory demands and move toward agreement by making concessions and searching for new alternatives [7].

This paper makes five main contributions towards the goal of developing intentional agents with negotiation competence. The first is to present a generic negotiation mechanism that handles multi-party, multiple-issue and single or repeated rounds. The second is to describe an approach for defining a structure for negotiation problems. The structure of a negotiation problem represents a natural link between planning and negotiation and allows the direct integration of these two cognitive capabilities into a control architecture for intentional agents. The third contribution is to describe a method for proposal generation that supports cooperative and non-cooperative negotiation behavior. The fourth contribution is to introduce a method for generating counterproposals without making concessions. The method, called bargaining issue manipulation, consists mainly of adding to a proposal negotiation issues considered superfluous, in the hope that the other parties will feel strongly about these issues. The last contribution is to describe a set of negotiation tactics for generating counterproposals, by making concessions.

Intentional agents equipped with the negotiation mechanism are currently being implemented in Prolog. In earlier work, we described intentional agents, defined the concept of conflict of interests, and presented part of the negotiation mechanism [5]. In this paper, we continue the description of the negotiation mechanism.

This paper is formatted in the following manner. Initially, we will describe a single intentional agent from an internal point of view. Next, we will present a detailed description of the four components of the negotiation mechanism outlined above, placing emphasis on bargaining issue manipulation and negotiation tactics. We then compare the negotiation mechanism with other developed mechanisms. Finally, we will present the concluding remarks and some important future avenues of research.
2. Intentional agents

Let \(\text{Agents} = \{a_{g_1}, \ldots, a_{g_r}\}, r\in\mathbb{N}\), be a set of intentional agents. This section presents a brief description of a single intentional agent \(a_{g_i}\in\text{Agents}\) (see our earlier work for an in-depth description [5]).

2.1. Mental state and mental attitudes

The mental state \(M_i\) of \(a_{g_i}\) is a 5-tuple: \(M_i = <B_i, G_i, PL_i, I_i, Ext_i>\), where \(B_i\) is a belief set, \(G_i\) a goal set, \(PL_i\) a plan template library, \(I_i\) an intention structure, and \(Ext_i\) an external description.

The agent \(a_{g_i}\) has a set \(B_i = \{b_{i1}, b_{i2}, \ldots\}\) of beliefs and a set \(G_i = \{g_{i1}, g_{i2}, \ldots\}\) of goals. Beliefs represent facts about the world and the agent himself. Goals represent world states to be achieved.

The agent \(a_{g_i}\) has a library \(PL_i = \{pt_{i1}, pt_{i2}, \ldots\}\) of plan templates representing known procedures for achieving specific goals. A plan template \(pt_{ik}\in PL_i, k\in\mathbb{N}\), is a 7-tuple: \(pt_{ik} = <id, name, args, type, body, constrs, preconds>\), where \(id\) is a plan identifier, \(name\) is a plan name, \(args\) is a list of arguments, \(type\) is the type of the plan template (composite or primitive), \(body\) is a procedure for achieving the goal specified by the name and args, \(constrs\) is a list of constraints (imposing a temporal order on the members of the body, etc), and \(preconds\) is a list of conditions that must hold before processing the plan’s body. The library \(PL_i = CPL_i \cup PPL_i\) has composite and primitive plan templates. A composite plan template \(cpt_{ij}\in CPL_i\) has a body containing one or more body steps describing the hierarchical decomposition of the goal specified by the name and args into more detailed subgoals. A primitive plan template \(ppt_{ik}\in PPL_i\) has a body containing an action (or a sequence of actions) directly executable by the agent.

The intention structure \(I_i = \{it_{i1}, \ldots, it_{in}\}, n\in\mathbb{N}\), consists of a list of intention threads. A single intention thread \(it_{ik}\), \(1\leq k\leq n\), is an agenda of hierarchically related plan templates. More specifically, each intention thread contains the plan \(pt_{ik}\) adopted by \(a_{g_i}\) for achieving the goal \(g_{ik}\in G_i\). This plan is constituted by intentions \((int_{ik1}, \ldots, int_{ik2}, \ldots)\).

The external description \(Ext_i = \{Ext_i(a_{g_1}), \ldots, Ext_i(a_{g_r})\}\) is a data structure where the information about the other agents \((a_{g_1}, \ldots, a_{g_r})\in\text{Agents}\) present in the environment is stored. A single entry

![Fig. 1. Agent architecture (part of plan generation)](image-url)

\(Ext_i(a_{g_i}) = <B_i(a_{g_i}), G_i(a_{g_i}), I_i(a_{g_i})>, 1\leq i\leq n\), contains the beliefs, goals and intentions (plans) that \(a_{g_i}\) believes \(a_{g_j}\) has.

2.2. Plan Generation

A plan \(p\) is a partially ordered collection of plan templates. The plan has a structure \(Pstruct\) consisting of a hierarchical and temporally constrained And-tree. The nodes of the tree are plan templates retrieved from the library and instantiated with specific values (henceforth, plan templates are referred as plans).

Figure 1 is a block diagram describing part of the architecture of an intentional agent, namely the part related to plan generation. At any given time, the agent \(a_{g_i}\) selects a goal \(g_{ik}\in G_i\) to achieve and starts the generation of the plan \(pt_{ik}\).

Plan retrieval consists of searching the plan library \(PL_i\) and finding any plan whose name and arguments match the description of \(g_{ik}\). When suitable matches are found, the plans are retrieved from the library and their arguments are unified with specific values from \(g_{ik}\) description. These plans are called retrieved plans.

Plan selection and adoption involves the evaluation of the retrieved plans \((pt_{i1}, \ldots, pt_{ik-1}, pt_{ik}, pt_{ik+1})\in PL_i\), the selection of the preferred one \(pt_{ik}\), and the adoption of \(pt_{ik}\). The evaluation and selection of the retrieved plans is done by comparing their costs (scores) and choosing the plan \(pt_{ik}\) with the higher cost [6]. The adoption of \(pt_{ik}\) consists of adding it to the intention structure \(I_i\). More specifically, \(pt_{ik}\) is added to the intention thread \(it_{ik}\). The description (name and args) of \(pt_{ik}\) is referred to intention

---

*Proceedings of the 11th International Workshop on Database and Expert Systems Applications (DEXA’00)*

0-7695-0680-1/00 $10.00 © 2000 IEEE
Plan interpretation consists mainly of selecting a composite plan for subsequent decomposition. Typically, the plan $p_{ik}$ recently added to $i_{ik}$ is a composite one and is selected.

Plan decomposition consists of processing the precondition list of the selected plan $p_{ik}$, and establishing a temporal order for its body members $\{g_{ik1}, g_{ik2}, \ldots\}$. The temporal order is defined by the constraint list. The body members $\{g_{ik1}, g_{ik2}, \ldots\}$ of $p_{ik}$ are interpreted as subgoals of $g_{ik}$ and are added to the goal set $G_{i}$ (assuming consistency is maintained). At this point a new cycle begins and all the tasks just described are repeated.

### 3. Multi-agent negotiation

Intentional agents operate in multi-agent systems and their plans often conflict with the plans of other agents. A conflict of interest is a social concept involving two or more agents that have plans requiring mutually exclusive world states to exist. These plans are called incompatible and cannot be executed together (see our earlier work for an in-depth description [5]).

Conflicts of interests raise negotiation problems. Let $A=\{ag_{i1}, \ldots, ag_{in}\}, A \subseteq Agents$, be a set of intentional agents. Let $P=\{p_{1k}, \ldots, p_{nk}\}$ be a set of plans of the agents in $A$. Let $ag_{i} \in Agents$ be an intentional agent. Let $p_{ik}$ be a plan of $ag_{i}$ incompatible with the plans in $P$. A negotiation problem from the perspective of the individual agent $ag_{i}$ is defined formally as a 4-tuple: $NP_{ik}=\langle ag_{i}, p_{ik}, A, P \rangle$.

#### 3.1. The generic negotiation mechanism

The mechanism defines the essential steps the agent $ag_{i}$ will perform in order to solve a negotiation problem $NP_{ik}$. The mechanism handles multi-party, multiple-issue and single or repeated rounds, and supports the following primary aspects of negotiation: (i) iterative exchange of proposals and counterproposals, (ii) application of non-concession and concession mechanisms for proposal and counterproposal generation, and (iii) learning in negotiation – discovery of new negotiation issues.

The mechanism is shown schematically in Fig. 2. First, $ag_{i}$ generates a structure $NPstruct_{ik}$ for the negotiation problem $NP_{ik}$ (see subsection 3.2). Next, he generates the preferred negotiation proposal $prop_{ikm}$ that satisfies the requirements imposed by $NPstruct_{ik}$ (see subsection 3.3). A negotiation proposal, broadly speaking, is a partial or a complete solution to the problem $NP_{ik}$ that $ag_{i}$ faces. Next, $ag_{i}$ determines the
feasibility of the proposal prop_{ikm}. If prop_{ikm} is acceptable, ag_i communicates it to the other parties (agents in A) which, typically, respond with counterproposals. A counterproposal is just a proposal made in response to a previous proposal prop_{ikm} and more favorable to the responding agent than prop_{ikm}.

Following this, ag_i evaluates the counterproposals and either accepts one of them or not. If all the counterproposals are unacceptable and the negotiation deadline is not reached, ag_i responds by sending back a counterproposal. He can generate a counterproposal by either: (i) “improving” the rejected proposal prop_{ikm} (see subsection 3.4), or (ii) preparing another proposal prop_{ikm+1} satisfying the requirements imposed by NPstruct_{ik} (see subsection 3.5). Both types of responses require no changes in NPstruct_{ik}.

The exchange of proposals and counterproposals continues until an agreement is found, or the negotiation deadline is reached, or the possibilities for “improving” rejected proposals and preparing new proposals have been exhausted, or a decision for introducing new issues has been taken. In the latter two cases, ag_i needs to restructure the negotiation problem. Problem restructuring allows the dynamic addition of negotiation issues and closes one round of negotiation. Negotiation proceeds to a new round in which all the tasks described above are repeated. The negotiation outcome is an agreement or a deadlock.

3.2. Negotiation problem structure generation

The negotiation problem NP_{ik} is represented by a hierarchical And-Or tree called negotiation problem structure (NPstruct_{ik}). The description (name and args) of the root node is called the negotiation goal (ng_{ik}) and the descriptions of the leaf nodes are called facts.

The negotiation problem structure NPstruct_{ik} is generated from the structure of plan pt_{ik} by expanding all the alternative plan templates. First, an initial structure is generated for NP_{ik}. This structure is simply a copy of pt_{ik}’s structure (And tree). Next, an iterative procedure involving five main processes starts. These processes are: (i) problem structure interpretation, (ii) plan decomposition, (iii) goal selection, (iv) plan retrieval, and (v) plan addition and placement.

Problem structure interpretation consists mainly of selecting an alternative plan pt_{ik} from the structure of NP_{ik}. Plan decomposition, goal selection and plan retrieval were described above (see subsection 2.2). In brief, these processes involve the definition of an order among the body members of pt_{ik}, the addition of the body members (goals) to a particular goal stack, the selection of a goal, and the retrieval of all the plan templates stored in the library PL_{ij} matching the goal description.

Plan addition and placement consists of placing all retrieved plan templates at appropriate points in the structure of NP_{ik}. The complete expansion of all the alternative plans leads to NPstruct_{ik}.

3.3. Proposal generation and evaluation

Let us introduce some formal definitions needed for describing proposal generation and evaluation. Let F_{ik}={f_{ik1}, ..., f_{ikz}}, z\in N, be the set of facts of NP_{ik}. A fact is defined formally as a 2-tuple: <issue, value>, where issue is a negotiation issue and value is a value for the issue. Facts can have logical values taken from the set L={true, false, any}. The set of issues under negotiation is represented by \( I_{ik} = \{i_{ik1}, ..., i_{ikj}\} \) and the generic value of the jth issue by \( x[i_{ikj}] \), 1\leq j\leq z. The issues are defined over finite ranges. The range of values acceptable to agent ag_i for the issue i_{ikj} is represented as \( D_{ikj} = [\min_{ikj}, \max_{ikj}] \). The agent assigns different weights to each issue reflecting their importance. The weight of the jth issue is represented by \( w_{ikj} \). We assume that the weights are normalized, i.e., \( \sum_{j=1}^{z} w_{ikj} = 1 \).

Proposal generation involves: (i) problem resolution (problem solution generation), (ii) position generation, (iii) position evaluation and selection, and (iv) proposal preparation.

The negotiation problem NP_{ik} has usually several solutions \( S_{ik} = \{sol_{ik1}, sol_{ik2}, \ldots \} \). A solution \( sol_{ikm} = (f_{ik1}, \ldots, f_{ikp}) \), sol_{ikm}\subseteq F_{ik}, is a set of facts with logical values satisfying the negotiation goal ng_{ik} [2]. The solution sol_{ikm} is obtained from NPstruct_{ik} by: (i) selecting exactly one disjunct from each Or node, and (ii) assigning logical values to the facts \( f_{ik1}, \ldots, f_{ikp} \) of the tree resulting from step (i).

This solution procedure partitions the set of facts \( F_{ik} \) of NP_{ik} into: (i) the subset I{F}_{ikm}={f_{ik1}, \ldots, f_{ikp}}, p\leq z, of the facts that constitute the solution sol_{ikm} and (ii) the subset FF_{ikm}={f_{ikp+1}, \ldots, f_{ikz}} of the remaining facts of NP_{ik} with I{F}_{ikm} \cap FF_{ikm} = \emptyset. The facts in I{F}_{ikm} are assumed to be non-negated and, therefore, they must...
have the logical value true in order to achieve \( n_{g_{ik}} \). They represent the inflexible facts of negotiation for the solution \( sol_{ikm} \). The issues \( INFissues_{ikm} = \{i_{is_{ik}}, \ldots, i_{is_{ikp}}\} \) associated with these facts represent the inflexible issues of negotiation.

A negotiation position \( pos_{ikm} = \{f_{ik1}, \ldots, f_{ikp}, f_{ikp+1}, \ldots, f_{ikz}\} \) is a solution \( sol_{ikm} = \{f_{ik1}, \ldots, f_{ikp}\} \) increased with the remaining facts \( FF_{ikm} = \{f_{ikp+1}, \ldots, f_{ikz}\} \) of \( NP_{ik} \). The facts in \( FF_{ikm} \) are not important for achieving \( n_{g_{ik}} \) and get the logical value any. They represent the flexible facts of negotiation for the solution \( sol_{ikm} \). The issues \( Fissues_{ikm} = \{i_{is_{ikp+1}}, \ldots, i_{is_{ikz}}\} \) associated with these facts represent the flexible issues of negotiation and are called bargaining issues.

The agent \( ag_i \) generates all the negotiation positions, determines their score using an additive scoring function [8], and selects the position with the higher score. Let \( V_{ikj} : D_{ikj} \rightarrow R \) be the component scoring function that gives the score that \( ag_i \) assigns to a value \( x(i_{is_{ikj}}) \in D_{ikj} \) of the generic issue \( i_{is_{ikj}} \in I_{ik} \). The score for contract \( x_{ikm} = (x(i_{is_{ik1}}), \ldots, x(i_{is_{ikp}})) \) corresponding to solution \( sol_{ikm} \) is given by:

\[
V_i(x_{ikm}) = \sum_{j=1}^{p} w_{ikj} V_{ikj}(x(i_{is_{ikj}}))
\]

The score for values \((x(i_{is_{ikp+1}}), \ldots, x(i_{is_{ikz}}))\) of the bargaining issues is assumed to be null. As a result, position \( pos_{ikm} \) has the same score as contract \( x_{ikm} \).

A negotiation proposal \( prop_{ikm} = \{f_{ik1}, \ldots, f_{ikp}, f_{ikp+1}, \ldots, f_{ikz}\} \) is a revised negotiation position obtained by changing the flexible facts with the value any to true or false. The agent \( ag_i \) selects the position \( pos_{ikm} \) with the higher score and changes the values of the facts in \( FF_{ikm} \) to true or false. He can adopt a cooperative or a non-cooperative behavior by assigning values to these facts according to or against the interests of other agents, respectively. He may also wish either to maintain all the flexible facts in the proposal or to withhold judgement on some of them. In the latter case, he selects a subset \( CFF_{ikm} = \{f_{ikp+1}, \ldots, f_{ikz}\} \). \( CFF_{ikm} \subseteq FF_{ikm} \) of the flexible facts and includes only these facts in the proposal \( prop_{ikm} = \{f_{ik1}, \ldots, f_{ikp}, f_{ikp+1}, \ldots, f_{ikz}\} \). This proposal is called a negotiation offer. The facts in \( CFF_{ikm} \) are called communicable flexible facts and the remaining facts, i.e., the facts in \( RFF_{ikm} = FF_{ikm} - CFF_{ikm} = \{f_{ikp+1}, \ldots, f_{ikz}\} \) are called retained flexible facts [2].

### 3.4. Counterproposal generation without concessions – bargaining issue manipulation

The agent \( ag_i \) communicates the negotiation proposal (or offer) \( prop_{ikm} \) to the other parties which, in turn, evaluate it and typically respond with counterproposals. Upon receiving the counterproposals, \( ag_i \) may decide either to accept one of them or to send back another counterproposal, i.e., an alternative proposal to \( prop_{ikm} \).

Counterproposal generation without restructuring the negotiation problem can be done by using: (i) non-concession, and (ii) concession mechanisms. Non-concession mechanisms generate counterproposals that have a score similar to the score of one or more previously offered counterproposals. This subsection introduces a significant non-concession mechanism that we call bargaining issue manipulation. Concession mechanisms will be the subject of the next subsection.

Successful real world negotiations often involve strategic misrepresentations. For instance, bargainers are often advised that they should purposely add to the negotiation agenda issues that they do not really care about, in the hope that the other parties will feel strongly about one or more of these issues - strong enough to be willing to make compensating concessions [8]. To this end, bargaining issue manipulation is a mechanism that allows one party to act strategically by exaggerating the importance of some issues (bargaining issues) in order to extract concessions from the other parties. More specifically, this mechanism allows one party \( ag_i \) to “improve” a rejected proposal \( prop_{ikm} \) by selecting one or more bargaining issues with specific values (flexible facts) from \( RFF_{ikm} \) and adding them to the proposal. The “improved” proposal \( prop_{ikm+1} \) and the rejected proposal \( prop_{ikm} \) have the same score (hence, \( ag_i \) does not make a concession).

Bargaining issue manipulation is formalized by two functions: select and add. The function select:

\[
\exists RFF_{ikm} \rightarrow RFF_{ikm} \text{ assists } ag_i \text{ in selecting a fact from } RFF_{ikm} \]

This function takes \( RFF_{ikm} \) as input and returns a retained flexible fact \( f_{ikx} \in RFF_{ikm} \). The function add:

\[
\exists prop_{ikm}, RFF_{ikm} \rightarrow prop_{ikm+1} \text{ maps a proposal } prop_{ikm} \text{ and a fact } f_{ikx} \in RFF_{ikm} \text{ into a new proposal } prop_{ikm+1} \text{ containing } f_{ikx}, \text{i.e.,}
\]

\[
\text{add}(prop_{ikm}, f_{ikx}) = prop_{ikm+1} \cdot f_{ikx}
\]
where \( \cdot \) stands for concatenation. It should be stressed that several select functions can be defined reflecting the negotiation behavior of \( ag_i \). One such function is:

\[
select(RFF_{ikm}) = f_{ik} \mid w_{ik} = \max_{q+1 \leq j \leq z} w_{ikj}
\]

where \( w_{ikj} \) \( (q+1) \leq j \leq z \), is the weight of the bargaining issues associated with the retained flexible facts. This function selects the retained flexible fact \( f_{ik} \) in \( RFF_{ikm} \) corresponding to the bargaining issue with the higher weight or importance.

### 3.5. Counterproposal generation with concessions—negotiation tactics

Concession mechanisms generate counterproposals that have successively lower scores than previous counterproposals. Negotiation tactics are the predominant concession mechanisms.

**Negotiation tactics** are functions that compute values for single issues at every instant in the negotiation. Let \( prop_{ikm} \) be a negotiation proposal submitted by \( ag_i \) to the other parties and rejected. Let \( x[is_{ikj}]_{old} \) be the value of the issue \( is_{ikj} \) offered in \( prop_{ikm} \) by \( ag_r \) and \( D_{ikj} \) be the range of values acceptable to \( ag_i \) for \( is_{ikj} \). Let \( V_{ikj} \) be the component scoring function of \( ag_i \) for issue \( is_{ikj} \). This function is assumed to be either monotonically increasing or monotonically decreasing. Let \( prop_{ikm+1} \) be the new proposal that \( ag_i \) decided to prepare as a response to the rejection of \( prop_{ikm} \). Generally speaking, a negotiation tactic is a function tactic: \( D_{ikj} \rightarrow D_{ikj} \), that takes the value \( x[is_{ikj}]_{old} \) of \( is_{ikj} \) as input and returns the new value \( x[is_{ikj}]_{new} \) of \( is_{ikj} \) to be offered in \( prop_{ikm+1} \) by \( ag_r \), with \( V_{ikj}(x[is_{ikj}]_{old}) \geq V_{ikj}(x[is_{ikj}]_{new}) \).

In this work, we consider five generic tactics that model different concessions on a single issue \( is_{ikj} \) at each point of the negotiation process. A concession on an issue \( is_{ikj} \) is defined as a change in the value of \( is_{ikj} \) that reduces the level of benefit sought (or score value). The description of the five tactics is as follows:

1. **Stalemate** - models a null concession on \( is_{ikj} \) i.e., the agent \( ag_i \) does not change the value of \( is_{ikj} \).
2. **Tough** - models a small concession on \( is_{ikj} \).
3. **Moderate** - models a moderate concession on \( is_{ikj} \).
4. **Soft** - models a large concession on \( is_{ikj} \).
5. **Compromise** - models a complete concession on \( is_{ikj} \) i.e., the new value \( x[is_{ikj}]_{new} \) of \( is_{ikj} \) is the value proposed by another agent for \( is_{ikj} \) (or the reservation value for \( is_{ikj} \) - see below).

Let us introduce some definitions needed to formalize these tactics. The **limit or reservation value** \( RV_{ikj} \) for issue \( is_{ikj} \) is the minimum (or maximum) value that \( ag_i \) is willing to accept before he definitely breaks off negotiation - depending on whether \( ag_i \) is maximizing or minimizing \( is_{ikj} \) [8]. The demand \( dem_{ikj} \) of \( ag_i \) for \( is_{ikj} \) is the value of \( is_{ikj} \) proposed by \( ag_i \) to the other parties at a specific point in the negotiation. The **negotiation interval** \( NI_{ikj} \) for \( is_{ikj} \) is the range of values between \( RV_{ikj} \) and \( dem_{ikj} \). Typically, \( NI_{ikj} \) diminishes during the course of negotiation [7, 8]. The new value \( x[is_{ikj}]_{new} \) of \( is_{ikj} \) is computed by the following expression:

\[
 x[is_{ikj}]_{new} = x[is_{ikj}]_{old} + (-1)^w \cdot F \cdot RV_{ikj} - x[is_{ikj}]_{old}
\]

where \( w = 0 \) if \( V_{ikj} \) is monotonically decreasing or \( w = 1 \) if \( V_{ikj} \) is monotonically increasing, and \( F \in [0, 1] \) is a factor. This expression assures that \( x[is_{ikj}]_{new} \leq RV_{ikj} \). It also models the following experimental conclusions [7]: (i) a higher limit should produce larger demands and slower concessions, and (ii) demand will be closer to limit the higher the limit. In addition, the expression assures that \( x[is_{ikj}]_{new} \geq x[is_{ikj}]_{old} \) i.e., once a value is offered for \( is_{ikj} \) it is not reversed. This is a basic principle of negotiation [7, 8] (tradeoffs and problem restructuring are the two notable exceptions - however, these mechanisms are distinct from negotiation tactics).

The factor \( F \) can be simply a constant [3]. The five tactics are then formalized by considering different values for \( F \). For instance, the stalemate tactic is formalized by setting \( F=0 \), the tough tactic by \( F \in [0, 0.5] \), the moderate tactic by setting \( F=0.5 \), the soft tactic by \( F \in [0.5, 1] \), and the compromise tactic by \( F=1 \) or \( F=\left[ x[is_{ikj}]_{old} \cdot x[is_{ikj}]_{old} \cdot \mid RV_{ikj} - x[is_{ikj}]_{old} \right] \), where \( x[is_{ikj}]_{old} \) is the value proposed by other party \( ag_r \) to the issue \( is_{ikj} \).

Alternatively, the factor \( F \) can vary throughout the negotiation and be a function of a single variable or criteria [1]. In this article, we concentrate on the relative concession criteria. Let \( x[is_{ikj}]_{1}, x[is_{ikj}]_{2}, \ldots, x[is_{ikj}]_{n} \), \( x[is_{ikj}]_{n} \) be the values of \( is_{ikj} \) successively offered by \( ag_i \) to the other parties, with \( V_{ikj}(x[is_{ikj}]_{i}) \geq V_{ikj}(x[is_{ikj}]_{j}) \), \( 1 \leq i, j \leq n \), and the compromise tactic by \( F=1 \) or \( F=\left[ x[is_{ikj}]_{old} \cdot x[is_{ikj}]_{old} \cdot \mid RV_{ikj} - x[is_{ikj}]_{old} \right] \), where \( x[is_{ikj}]_{old} \) is the value proposed by other party \( ag_r \) to the issue \( is_{ikj} \).
of the total concession made by $a_{i/k}$ on $is_{i/kj}$. We distinguish two functions for modelling $F$:

$$(1) \ F = 1 - \gamma \ e^{\gamma e^{\left|\text{CTotal}\right|}}$$

and

$$(2) \ F = \gamma e^{\left(1 - \left|\text{CTotal}\right|\right)}$$

where $\gamma \in \mathbb{R}^+$. These functions model the typical monotone decreasing pattern of concessions - the value of concessions becomes successively smaller as the negotiators move closer to their limit [8]. They also model the typical shape of the demand curve over time - demand declines rapidly at first and then more and more slowly as time goes on [7]. The five negotiation tactics are now formalized by choosing different values for the parameter $\gamma$. For function (1), the stalemate tactic is formalized by setting $\gamma \epsilon \left|\text{CTotal}\right|$, the tough tactic by setting $\gamma \epsilon 1$, the moderate tactic by setting $\gamma = 1$, the soft tactic by $\gamma \epsilon 0$, and the compromise tactic by $\gamma = 0$ or $\gamma = \left(\text{RV}_{i/kj}^{x} (\text{is}_{i/kj}) \right)^{\gamma} e^{\left|\text{CTotal}\right|}$.

4. Related work

Negotiation is a rich, multidisciplinary research area. As a result, we highlight in this section just the negotiation work most related to our own work. Laasri et al. [4] present a generic negotiation mechanism. The mechanism is rich, but assumes that agents are inherently cooperative. Sycara [9] presents a negotiation mechanism that can be employed by non-cooperative agents and supports problem restructuring. However, the mechanism assumes the existence of a centralized mediator. Faratin et al. [1] present a multi-party, multi-issue, single encounter negotiation mechanism. Again, the mechanism is rich, but no consideration was given to integrate it in a complete agent architecture.

We are interested in negotiation among self-motivated agents. Our structure for representing negotiation problems allows the direct integration of planning and negotiation into a complete agent architecture. This structure is similar to goal representation trees [2], but there are important differences. Our approach is based on plan templates and plan expansion and not on production rules and forward or backward chaining. Also, our formula for modelling negotiation tactics is similar to the formulae used by Faratin et al. [1] and Koperacz et al. [3]. Again, there are important differences. Our formula assures that the new value of an issue ranges between the reservation value and the previous value of the issue. In addition, our formula models important experimental conclusions about limit, demand, and concession.

Finally, the relative concession criteria is not used by other researchers.

5. Discussion and future work

This article has introduced a negotiation mechanism, and methods for negotiation problem structuring, proposal generation, and counterproposal generation either with and without making concessions. There are several features of our work that should be highlighted. First, the negotiation mechanism is generic and can be used in a wide range of domains. Secondly, the mechanism supports problem restructuring, ensuring a high degree of flexibility. Thirdly, the structure of a negotiation problem represents a natural link between the individual and social behavior of agents. This structure defines the set of negotiation issues. Also, problem structure generation acknowledges the role of conflict as a driving force for negotiation. Finally, proposal generation supports cooperative and non-cooperative negotiation behavior.

Our aim for the future is: (i) to extend the negotiation mechanism to consider proposal feasibility and problem restructuring, and (ii) to validate experimentally the mechanism.

6. References


