

Interest-Based Negotiation as an Extension of Monotonic Bargaining in 3APL

Philippe Pasquier¹, Frank Dignum³, Iyad Rahwan^{2,4}, and Liz Sonenberg¹

¹ University of Melbourne, Australia

² British University of Dubai, UAE

³ Utrecht University, The Netherlands

⁴ (Fellow) University of Edinburgh, UK

Abstract. Reframing is a sub-type of interest-based negotiation strategy that enhances bargaining by allowing the negotiators to ask for the underlying goal of the negotiation and propose alternative plan(s) which may entail a deal on alternative issues. This paper (i) presents a negotiation protocol that support both alternate offers monotonic bargaining and reframing and (ii) gives a fully computational and reproducible specification of bargaining and reframing capable negotiation agents using the 3APL agent language.

Keywords: Conflict resolution and negotiation, Dialog and interaction protocols, Agent programming languages, frameworks, and toolkits, Agent-oriented software engineering, Agent-based electronic commerce

1 Introduction

Argumentation-based negotiation (ABN) [10] has been proposed as a way to enhance classical negotiation dialogues between artificial agents and progress toward human computer negotiation. Interest-based negotiation (IBN) [9] is the subset of ABN that focusses on the argumentation about the interests underlying the negotiation. However, no work has proposed a concrete – computational and fully reproducible⁵ – way to implement IBN.

This paper advances the state of the art by addressing the software engineering level of negotiation agents. It shows how to develop software agents capable of bargaining and reframing, i.e. a sub-type of IBN. The proposed negotiation model consists of:

1. *The negotiation strategies*: an agent strategy is a sequence of actions, which in the context of negotiation mainly consists of offers and responses. Section 2 shows how reframing can be used to extend classic bargaining strategies. The set of possible strategies are constrained by the chosen protocol.
2. *The negotiation protocol*: the protocol specifies the rules of encounter between the negotiators. It defines what are the locutions and their possible sequences. Section 3 specifies the proposed protocol for alternate offers negotiation and reframing.

⁵ We refer the interested reader to [11] for an example of discussion and experiments showing that formal models usually failed to be reproducible computational characterization when not expressed directly in a particular programming environment.

3. *The information state of agents*: the agents' models reify the negotiation strategy and the negotiation protocol thus providing distributed decision making mechanisms suited to MAS. In our case, we use the 3APL agent framework (presented in Section 4) to specify reframing-capable cognitive agents (Section 5). Because 3APL is a declarative environment, those specifications are also the actual code for the agent programs.

Finally, Section 6 concludes by discussing the results associated with this model and its implementation.

2 Negotiation Strategy: From Bargaining to IBN

The challenge of negotiation is to allocate scarce resources to self interested parties. The resources can be quite anything as long as they are scarce, i.e. there is not enough to allow for an allocation that would maximally satisfy all parties. The object of negotiation may be tangible (bandwidth, money, property, processing power, ...) or intangible (better communication, better work performance, more respect, ...). The traditional form of negotiation, characterized by the assertion of opposing positions by the parties, is referred to as position-based negotiation, which in the context of e-business and economic exchanges is addressed as *bargaining*. This tends to view the object of the negotiation as fixed, such that a better deal for one means a lesser beneficial deal for the other; a "zero-sum game".

Bargaining strategies and mechanisms have been well explored both in multi-agent systems and economics [5, 8]. One of the properties, which is commonly agreed on in the analysis of bargaining strategies is the monotonicity of the bargainers' offers, i.e. buyers/sellers may only insist on their previous offers or raise/reduce their offers monotonically until an agreement is reached. Time dependant tactics [6], behavior dependent tactics [5] and market driven strategies [12] all have this monotonic offers property. While recent works have been looking at non-monotonic-offers bargaining protocols [14], we will only consider monotonic offers in this paper.

Originally developed for human negotiation and mediation practices and first introduced in [7], the theory of *interest-based negotiation* (IBN) shows that parties are much more likely to come to a mutually satisfactory outcome when the object of the negotiation is not considered as central as the agents' underlying interests. By focusing on interests to be satisfied rather than positions to be won, IBN allows the agents to search the space of negotiation objects (rather than the space of deals for a particular item). When successful, this strategy gives each side more, thereby producing a "win-win" outcome.

IBN has been adapted and applied to MAS negotiation dialogues [9]. In that context, it is a subclass of argumentation-based negotiation (where the agents are arguing about negotiation related issues, i.e. beliefs, goals or social aspects). Interest-based negotiation rests on the idea that the agents can explicit the goals underlying the negotiation and discuss alternative ways to achieve these.

While classical negotiation models (heuristic approaches or game theoretic approaches) focus on processes to accommodate the agents' preferences, in IBN,

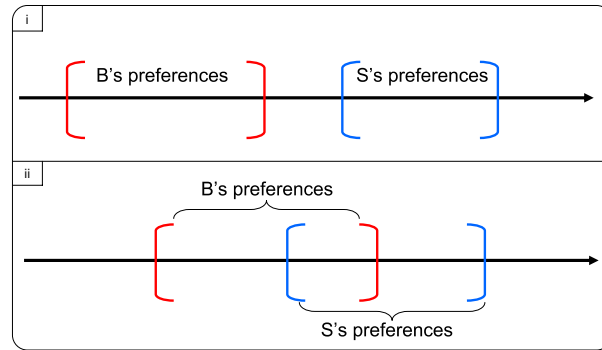


Fig. 1. Upper and lower bounds on agent preferences ordering.

the agents' preferences may change. Let's look at some informal examples to illustrate this idea.

First, consider the following bargaining dialogue where the agents' preferences are summarized by Figure 1, part (i). In that example, the agents fail to reach a deal because it is not possible to accommodate their respective preferences.

B₁: I would like to rent a car for 4 days please.
S₁: I offer you one for \$400.
B₂: I reject! How about \$200?
S₂: I reject!

Of course, sometimes, the agents preferences overlap (as in Figure 1, part (ii)) and a deal can be reached using a monotonic bargaining strategy like in the following dialogue:

B₁: I would like to rent a car for 4 days please.
S₁: I offer you one for \$400.
B₂: I reject! How about \$200?
S₃: I reject! How about \$300 then?
B₃: I guess that's the best I can do, I accept!

However, it is worth noticing that agents had to make concessions and move away from their preferred outcome in order to reach a deal. The following dialogue gives an example of IBN dialogue where the seller agent (*S*) asks the buyer to give his underlying interest (goal) before making a concession. In this example, the seller knows an alternative way to fulfill the buyer's underlying goal. Assuming that the seller earns a \$100 profit on both the rent of a car for \$400 and a \$200 Quantum ticket, an agreement is reached without any concession.

B₁: I would like to rent a car for 4 days please.
S₁: I offer you one for \$400.
B₂: I reject! How about \$200?
S₂: I reject! What do you need a car for?
B₃: I want to drive to Sydney to attend a conference.
S₃: You can also fly to Sydney! I can book you a ticket with Quantum airlines for \$200.
S₄: I didn't know flights were so cheap! I accept!

This particular negotiation strategy is refereed in the human negotiation literature as *reframing*. The two main strategic issues to model when one want to use reframing in order to enhance bargaining-based negotiation are:

- The bargaining function used by the agent to compute their next offers. While complex bargaining functions are available in the literature, we use a simple but standard one here. Starting with initial offers that are their most preferred options, the agents will use the mean ($\frac{X+Y}{2}$) of the participating agents current offers (X and Y) as long as it falls within the preferences boundaries.
- When to stop bargaining and try reframing? In order to embed IBN in negotiation dialogues, there must be a point where bargaining stops. This point does not have to be the same as when one would only have bargaining available as a strategy. In this paper, reframing is triggered when both agents repeat their offers, which indicates that they reached their less preferred limit and that the bargaining failed.

While the choices made in this paper at these strategic levels are very simple, complex functions could be devised to make the decision based on knowledge of opponent, predictions based on bargaining history, costs of information disclosure, ... The rest of this paper presents and discusses an implementation of software agents capable of monotonic bargaining and reframing.

3 The Negotiation Protocol

In order to enable agents to use reframing, one needs to define an appropriate protocol that allows both bargaining and reframing. As advocated in [13], we break the protocol into two parts, making it more modular. Figure 2 thus presents the UML 2.0 specification of the two parts of the protocol: (1) the bargaining protocol (on the left) and (2) the reframing one (on the right).

The bargaining protocol is a classical alternated offers protocol. It is made of the sequencing of the initiator's request, and the subsequent refusal or proposal from the partner. Then the protocol allows alternated counter proposals from both parties with possible refusal or acceptance at each stage. Original to this protocol is the possibility to embed a reframing dialog instead of making a proposal (or counter proposal).

The reframing protocol mainly consists in allowing its initiator to request the partner for the underlying goal of the negotiation. The partner can inform

the initiator of this or refuse to answer. In the case where the partner informs the bidder of his underlying goal, the initiator can be clueless and quit the protocol with a cancel message or he can inform the partner of an alternative plan for achieving the goal. Reframing, as a strategy, rests on the hypothetical existence of alternative plan(s) to achieve the underlying goal of the initiator involving different resources for which bargaining will be more advantageous for both parties.

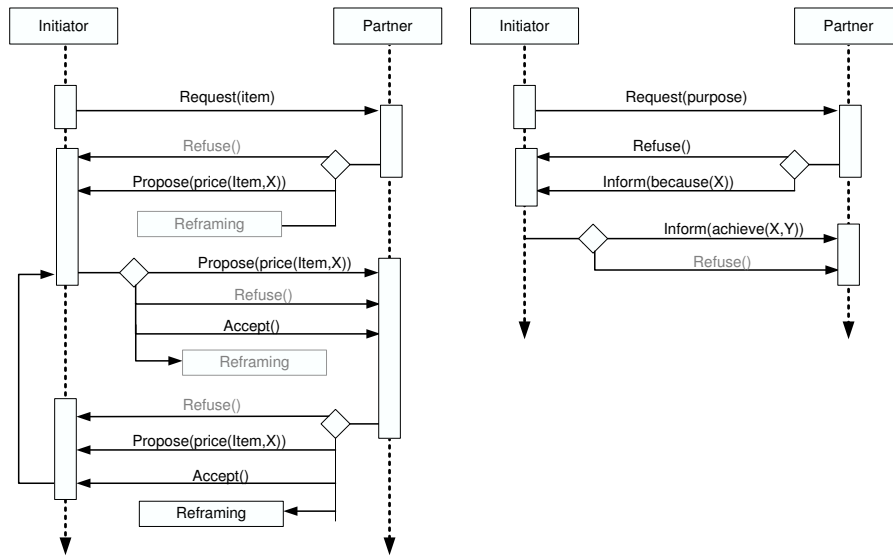


Fig. 2. UML 2.0 specification of the proposed bargaining and reframing protocols.

4 Agents' information states: the 3APL agent platform

In order to give a fully reproducible computational characterization of the ideas developed here, we will use the 3APL agent language to present our model. Fortunately, the 3APL language has been developed precisely to help bridging the gap between theoretical models (in particular, logic-based) and practical implementation of MAS concepts.

Since 3APL is well documented and discussed in [4, 3, 2], we only sum up the basics here. A 3APL agent is made of 6 bases:

1. A *beliefs base*: the agent belief base is expressed as a Prolog⁶ code (ISO standard).⁷ External Prolog programs can be embedded thus allowing to

⁶ For space reasons, we don't detail the Prolog syntax in here. Suffice to remember that variables are noted with capitalized identifiers.

⁷ 3APL use the JIP Prolog java package, see [1] for more information.

```

<Program> ::=      "Program" <ident>
                  ( "Load" <ident> )?
                  "Capabilities:" ( <capabilities> )?
                  "BeliefBase:" ( <beliefs> )?
                  "GoalBase:" ( <goals> )?
                  "PlanBase:" ( <plans> )?
                  "PG-rules:" ( <p_rules> )?
                  "PR-rules:" ( <r_rules> )?

<capabilities> ::= <capability> ( ";" <capability> )*
<capability> ::=  "{" <query> "}" <Atom> "{" <literals> "}"
<beliefs> ::=    ( <belief> )*
<belief> ::=     <ground_atom> ":", " | <atom> ":" -" <literals> ":"
<goals> ::=      <goal> ( ";" <goal> )*
<goal> ::=       <ground_atom> ( "and" <ground_atom> )*
<plans> ::=      <plan> ( ";" <plan> )*
<plan> ::=       <basicaction> | <composedplan>
<basicaction> ::= "ε" | <Atom> | "Send("<iv>,<iv>,<atom>")" |
                  "Java("<ident>,<atom>,<var>")" | <wff> "?" | <atom>
<composedplan> ::= "if" <wff> "then" <plan> ( "else" <plan> )? |
                  "while" <query> "do" <plan> |
                  <plan> ";" <plan>
<p_rules> ::=     <p_rule> ( ";" <p_rule> )*
<p_rule> ::=      <atom> "<-" <query> "|" <plan>
<p_rule> ::=      "<-" <query> "|" <plan>
<r_rules> ::=     <r_rule> ( ";" <r_rule> )*
<r_rule> ::=      <plan> "<-" <query> "|" <plan>
<literals> ::=    <literal> ( ";" <literal> )*
<literal> ::=     <atom> | "not("&<atom>")"
<wff> ::=        <literal> | <wff> "and" <wff> | <wff> "or" <wff>
<query> ::=       <wff> | "true"
<iv> ::=          <ident> | <var>

```

Fig. 3. The EBNF specification of the 3APL language for programming individual agents where: $\langle atom \rangle$ refers to Prolog ground atomic formulas (can include Prolog-like list representation), $\langle Atom \rangle$ denotes atomic action formulas (starting with a capital letter), $\langle ident \rangle$ denotes strings and $\langle var \rangle$ denotes variables (starting with a capital letter as in Prolog).

give common knowledge to the agents and outsource more complex reasoning capabilities;

2. A *capabilities base*: the capabilities base stores all the mental and external parameterized actions known by the agent. All capabilities are expressed as: $\{\langle Precondition \rangle\} \langle Action(parameters) \rangle \{\langle Effects \rangle\}$, where $\langle Precondition \rangle$ is interpreted as a query to the belief base and $\langle Effects \rangle$ captures the effects of the action $\langle Action() \rangle$ on the agent's belief base, in terms of addition and deletion of beliefs;
3. A *goal base*: this base contains the agent's goal(s) expressed as grounded Prolog expressions;
4. A *plan base*: the plan base holds the agent's current plans expressed in terms of actions and (sub)plans, using the following planing operators: sequence (;), conditional (*if* $\langle query \rangle$ *then* $\langle plan \rangle$ *else* $\langle plan \rangle$) and iteration (*While* $\langle query \rangle$ *do* $\langle plan \rangle$);
5. A *goal planning rules base*: the goal planing rules are used to generate plans to achieve goals. These are expressed as: $\langle head \rangle \leftarrow \langle guard \rangle \mid \langle body \rangle$ where

$\langle head \rangle$ is a goal, $\langle guard \rangle$ is belief query that checks if the rule can be applied and $\langle body \rangle$ is the plan to execute;

6. A *plan revision base*: the plan revision rules are used to revise plans from the plan base. These are also expressed in terms of head (plan), guard (belief query) and body (plan). The head of the rule is replaced by the body if the guard holds. The rule thus changes the plan and no backtracking is possible;

Furthermore, 3APL includes FIPA compliant communication facilities through the special and built in action $Send(\langle interlocutor \rangle, \langle performative \rangle, \langle content \rangle)$. Such action adds the associated $sent(\langle interlocutor \rangle, \langle performative \rangle, \langle content \rangle)$ and $received(\langle sender \rangle, \langle performative \rangle, \langle content \rangle)$ beliefs in the interlocutor and sender belief bases respectively.

The EBNF specification of the 3APL language is given by Figure 3. All those structures are used and manipulated through the deliberation cycle of 3APL agents which is illustrated in Figure 4. The 3APL agent platform is made of an interface that allows defining and executing 3APL agents. Facilities like an agent management system (AMS) and a client server distribution are built-in.⁸

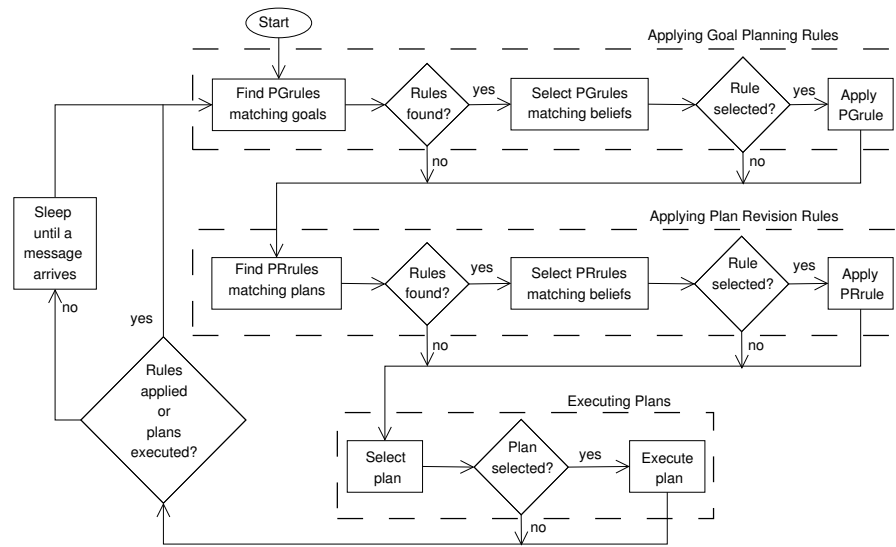


Fig. 4. 3APL agent deliberation cycle.

5 Implementing Reframing in 3APL: Detailed Example

In this Section, we give the specification-code of generic buyer and seller agents through a concrete example that can easily be adapted to a great number of

⁸ The 3APL distribution is available for download at: <http://www.cs.uu.nl/3apl/>.

actual negotiation contexts (e.g. in electronic commerce). Let us consider two agents Vero and Philippe whose specification-codes are indicated by Figures 5 and 6 respectively.

PROGRAM "Vero"

```

1: CAPABILITIES {
2: {True} Goal(Y) {goal(Y)}
3: {True} Treatedgo(X) {go(X)}
4: {True} Treated(X,C,Y) {not received(X,propose,content(price(C,Y)))}
5: {True} Updateoffer(Z,W,N) {not currentoffer(Z), not lastoffer(W), lastoffer(Z),
currentoffer(N)}
6: {True} Updateotheroffer(X,Y) {not otheroffer(X), otheroffer(Y)}
7: {True} Updatecheaper(Y,V) {not cheaper(V,Y), cheaper(Y,V)}
8: {True} Pay(X,Y) {payed(X,Y)}
9: }
10: BELIEFBASE {
11: transport(car). transport(plane). transport(boat). transport(train).
12: conf(aaai). conf(primus). conf (uumus).
13: town(sydney). town(candera). town (melbourne).
14: isin(aaai,sydney). isin(primus, candera). isin(uumus, melbounre)
15: pricereg(aaai,125). pricereg(primus, 75). pricereg(uumus, 200).
16: seller(philippe). seller(sally). seller(bob).
17: achieve(go(Z),W):- town(Z), transport(W).
18: cheaper(rentcar,plane). cheaper(rentcar,train). cheaper(rentcar,boat).
19: cheapest(X):- not cheaper(Y,X).
20: deal(price(C,Z)):- sent(X,accept,content(price(C,Z))), seller(X).
21: deal(price(C,Z)):- received(X,accept,content(price(C,Z))), seller(X).
22: securetransport(X):- deal(price(Y,Z)), transport(Y).
23: prefmin(rentcar,250). prefmax(rentcar,100).
24: prefmin(plane,250). prefmax(plane,150).
25: nextoffer(N):- currentoffer(Z), otheroffer(O), plus(Z,O,P), times(P,0.5,N).
26: currentoffer(0). lastoffer(0). otheroffer(0).
27: }
28: GOALBASE {
29: go(aaai) }
30: PLANBASE {}
31: PG-RULES {
32: go(X)← conf(X) and isin(X,Y) | {Goal(go(Y)); goal(go(Y)); payreg(X); Treatedgo(X)},
33: }
34: PR-RULES {
35: goal(X)← achieve(X,Z) and cheapest(Z) | negotiate(Z)
36: payreg(X)← pricereg(X,Y) and securetransport(X) | {Pay(X,Y)} }
37: negotiate(Z) ← seller(X) | {Send(X,request,content(Z)); bargain(X,Z)}
38: bargain(X,Content) ← received(X,propose,content(price(Content,Y))) and prefmin(Content,M)
and prefmax(Content,P) and otheroffer(T) and currentoffer(C) | {Treated(X,Content,Y);
Updateotheroffer(T,Y); minus(Y,C,D)?; if D < 10 then
Send(X,accept,content(price(Content,Y))) else {nextoffer(X,Content); bargain(X,Content)}},
39: bargain(X,Y) ← received(X,reqpurpose,why(Y)) and achieve(Z,Y) and goal(Z) and not
sent(X,inform,because(Z)) | {Send(X,inform,because(Z));reframe(Z,Y)},
40: bargain(X,Y) ← received(X,reqpurpose,why(Y)) and achieve(Z,Y) and goal(Z) and
sent(X,inform,because(Z)) | {Send(X,refuse,because(Z))},
41: bargain(X,Y) ← received(X,accept,content(price(Y,Z))) | {Treated(X,Y,Z)},
42: reframe(Z,V) ← received(X,inform,content(achieve(Z,Y))) and goal(Z) and not cheaper(Y,V) |
{Updatecheaper(Y,V); goal(Z)},
43: nextoffer(Y,X)← prefmin(X,M) and prefmax(X,A) and currentoffer(Z) and lastoffer(W) | if not
sent(Y,propose,content(price(X,A))) then {Send(Y,propose,content(price(X,A)))};
Updateoffer(Z,W,A) else nextoffer(N)?; if N > M then {Send(Y,propose,content(price(X,M)))};
Updateoffer(Z,W,M) else {Send(Y,propose,content(price(X,N))}; Updateoffer(Z,W,N)}}
44: }

```

Fig. 5. 3APL buyer program.

Vero has a number of beliefs about transport means (line 11), conferences (line 12), towns (line 13), conferences locations (line 14), conferences prices (line 15) and seller agents (line 16) (for space reason, we gave this agent knowledge about seller agents rather than having her discover them using 3APL’s AMS). She believes she can go to a town using any of the transport means (line 17) and that a deal is made whenever she or the seller accepts an offer (lines 20 and 21). She believes that she succeeds in securing some transportation when she has made a deal on a particular transport means (line 22). She has preferences’ boundaries over the prices of different items (lines 23 and 24). Vero also has a priori beliefs about the relative price of transport means (line 18) and knows that the cheapest would be the one(s) that do not have cheaper alternatives (line 19). Finally, the predicate *nextoffer()* captures the bargaining strategy function (line 25, as described in Section 2) and various variables are initialized (line 26).

Vero’s initial goal is to attend to a conference named *aaai* (line 29). She knows that in order to go to a conference she needs to go to the city where the conference is held and pay the registration (line 32). In order to pay the registration, she needs to be aware of the price of the registration and make sure that she has already secured transportation to the place where the conference occurs (line 36). Vero also knows that in order to achieve a goal she will have to negotiate on the cheapest way to accomplish it (line 35). Negotiation starts by sending a request for the wanted item to a seller and then bargain (line 37).

Bargaining is defined recursively using plan reasoning rules. On the reception of a proposal, she accepts it if it is sufficiently close to her own proposal or else, she makes a counter offer and the bargaining continues (line 38). The bargaining stops when she issues an acceptance message (line 38) or when she receives one (line 41). Her strategy for computing the next offer (line 43) matches the bargaining strategy discussed in Section 2, i.e. starting with the maximally preferred offer and increases monotonically up to the less preferred but still acceptable proposal.

Reframing occurs when she receives a request for purpose, which she answers by refusing it (line 40) or by informing the seller about her purpose (lines 39) and prepares for a reframing. If the partner informs her of a valuable alternative to achieve her underlying goal, she will adopt it as a subgoal and will start a new bargaining on it (line 42).

Philippe is a seller agent that maintains prices preferences’ boundaries upon the different items he is selling (lines 7 and 8, in Figure 6). His beliefs include the predicate *nextoffer* (line 10) which encode its bargaining strategy and *reframecondition* (line 11) which gives the condition for attempting a reframing (as described in Section 2). The protocol described in Section 3 is implemented in a generic way through 6 rules⁹:

1. when he receives a request for an item that has not been treated, he makes an initial proposal (his most preferred price) and enters a bargaining plan (line 17);

⁹ Note that the parts in grey in Figure 2 are not shown in the agents code for space reason.

PROGRAM "Philippe"

```
1: CAPABILITIES {
2: {True} Treated(X,C,Y) { not received(X,propose,content(price(C,Y)))}
3: {True} Updateotheroffer(Z,X,Y) {not lastotheroffer(Z), lastotheroffer(X), not otheroffer(X),
  otheroffer(Y)}
4: {True} Updateoffer(X,Y,Z) {not currentoffer(X), not lastoffer(Y), lastoffer(X), currentoffer(Z)}
5: }
6: BELIEFBASE {
7: prefmax(rentcar,500). prefmin(rentcar,300).
8: prefmax(plane,400). prefmin(plane,200).
9: currentoffer(0). otheroffer(0). lastoffer(0). lastotheroffer(0).
10: nextoffer(N):- currentoffer(Z), otheroffer(O), plus(Z,O,P), times(P,0.5,N).
11: reframecondition(X) :- currentoffer(X), lastoffer(X), otheroffer(Z), lastotheroffer(Z).
12: achieve(plane,go(sydney)).
13: }
14: GOALBASE { }
15: PLANBASE {}
16: PG-RULES {
17: ← received(X,request,content(Content)) and not sent(X,propose,content(price(Content,W)))
  and prefmax(Content,Y) and currentoffer(W) and lastoffer(L) | {Updateoffer(L,W,Y);
  Send(X,propose,content(price(Content,Y))); bargain(X,Content)}
18: }
19: PR-RULES {
20: bargain(X,Content) ← received(X,request,content(Content)) and
  received(X,propose,content(price(Content,Y))) and otheroffer(W) and currentoffer(Z) and
  lastotheroffer(P) | {Treated(X, Content, Y); Updateotheroffer(P,W,Y); minus(Z,Y,R)?; If R <
  10 then Send(X,accept,content(price(Content,Y))) else nextoffer(X,Content)},
21: bargain(X,Content) ← received(X,accept,content(price(Content,Y))) |
  {Treated(X,Content,Y)},
22: reframe(X,Content) ← not sent(X,reqpurpose,why(Content)) |
  {Send(X,reqpurpose,why(Content)); reframe(X,Content)} ,
23: reframe(X,Content) ← received(X,inform,because(Y)) and achieve(P,Y) and not
  sent(X,inform,content(achieve(Y,P))) | {Send(X,inform,content(achieve(Y,P)))} ,
24: nextoffer(X,Request)← prefmin(Request,M) and currentoffer(Z) and lastoffer(W) |
  {nextoffer(N)?; if N > M then {Updateoffer(Z,W,N);
  Send(X,propose,content(price(Request,N))); bargain(X,Request)} else { Updateoffer(Z,W,M);
  if not reframecondition(M) then {Send(X,propose,content(price(Request,M)));
  bargain(X,Request)} else reframe(X,Request) } } }
25: }
```

Fig. 6. 3APL seller program.

2. on the reception of a counter proposal, he can accept it if it is close enough to its current offer or make another offer (line 20);
3. the bargaining succeeds if he send an acceptance or if he receives ones (lines 20 and 21);
4. when he wants to make another offer, Philippe calculates his next proposal and checks if the reframing condition holds to decide whether to continue the bargaining or enter a reframing strategy (line 24);
5. reframing consists in asking the buyer for his underlying purpose (line 22);
6. if informed by the buyer of her underlying goal, reframing also involves sending him information about an alternative mean to achieve that goal, if any (line 23).

In this context, Vero's initial deliberation and planning on how to achieve her goal ends up in a request for a proposal about the rent of a car from one of the sellers she knows (Philippe is the first in her list). The agents bargain until Philippe asks for the underlying purpose of the initial request. Vero informs him that her goal is to go to Sydney. Philippe then informs her of an alternative way

Sender	Receiver	Performative	Content
vero	philippe	request	content(rentcar)
philippe	vero	propose	content(price(rentcar,500))
vero	philippe	propose	content(price(rentcar,100))
philippe	vero	propose	content(price(rentcar,300))
vero	philippe	propose	content(price(rentcar,200))
philippe	vero	propose	content(price(rentcar,300))
vero	philippe	propose	content(price(rentcar,250))
philippe	vero	propose	content(price(rentcar,300))
vero	philippe	propose	content(price(rentcar,250))
philippe	vero	reqpurpose	why(rentcar)
vero	philippe	inform	because(go(sydney))
philippe	vero	inform	content(achieve(go(sydney),plane))
vero	philippe	request	content(plane)
philippe	vero	propose	content(price(plane,400))
vero	philippe	propose	content(price(plane,150))
philippe	vero	propose	content(price(plane,275))
vero	philippe	propose	content(price(plane,212.5))
philippe	vero	propose	content(price(plane,243.75))
vero	philippe	propose	content(price(plane,228.125))
philippe	vero	propose	content(price(plane,235.9375))
vero	philippe	accept	content(price(plane,235.9375))

Fig. 7. Conversation between Philippe and Vero as generated in the 3APL communication viewer by the piece of code given in this paper (thus fully reproducible).

to achieve that goal, namely by taking a plane. Notice that because the goal base and the planning rules base of 3APL agents do not support update in the present implementation, we had to encode plans and (sub)goals as beliefs (through the predicates *achieve* and *goal* respectively).¹⁰ Once Vero is aware that she can achieve her goal to go to Sydney by booking a plane she issues a request for proposal for that new solution. The agents then start a new bargaining on that item and a deal is concluded. Figure 7 indicates the actual message exchange that occurs between the two 3APL agents when running them.

Note that if one changes the preferences of the agents about the price of a car rental so that they intersect, then the first bargaining dialogue would have been concluded without entering IBN, as expected with the chosen strategy.

6 Conclusion and discussion

In this paper, we introduce reframing, which is the most common interest-based negotiation strategy. This paper advances the state of the art by providing a ready-to-use specification of reframing as a way to enhance a monotonic bargaining negotiation strategy in case of failure. The benefit of reframing is to eventually offer an alternative to simple monotonic bargaining when it fails. In that sense, bargaining enhanced by reframing is a dominant strategy over pure

¹⁰ Another way to proceed would have been to have the rule in Vero’s planning rules initial base with a guard that checks if Vero is in the state of having knowledge of that rule. That solution would have been less realistic, since it would have assumed that Vero is implicitly aware of every information she can receive.

bargaining, i.e. no agent loose anything by trying it and they may gain substantial benefits.

In addressing the software engineering level of negotiating agents in a symbolic environment, we raise some limitations of the 3APL language, namely the fact that the goal base, the planing goal rules and plan reasoning rules are statically defined (one can't add some planning knowledge dynamically, e.g. as a result of communication on know-how). Ongoing and future work involves collaboration to extend 3APL in order to allow for dynamic updates of 3APL agents planning rules as well as as studying and modeling IBN strategies that are not limited to reframing.

References

1. U. Chirico. *JIP Prolog: Java Internet Prolog*.
2. M. Dastani, M. B. van Riemsdijk, and J.-J. Meyer. *Multi-Agent Programming: Languages, Platforms and Applications*, chapter Programming multi-agent systems in 3APL, pages 10–45. Springer-Verlag, 2005.
3. M. Dastani, J. vander Ham, and F. Dignum. Communication for goal directed agents. In M.-P. Huget, editor, *Communication in Multiagent Systems - Agent Communication Languages and Conversation Policies*, LNCS, pages 239–252. Springer-Verlag, 2003.
4. M. Datani, F. de Boer, F. dignum, and J.-J. Meyer. Programming agent delib-eration: An approach illustrated using the 3apl language. In *Proceedings of The Second Conference on Autonomous Agents and Multi-agent Systems (AAMAS'03)*, pages 97–104, 2003.
5. P. Faratin, C. Sierra, and N.R. Jennings. Negotiation decision functions for au-tonomous agents. *International Journal of robotics and Autonomous Systems*, 3-4(24):159–182, 1998.
6. S. S. Fatima, M. Wooldridge, and N. R. Jennings. Optimal negotiation strategies for agents with incomplete information. In *ATAL'01*, pages 53–68, Seattle, US, 2001.
7. R. Fisher and W. Ury. *Getting to Yes: Negotiating Agreement Without Giving In*. New York: Penguin Books, 1983.
8. N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parson, C. Sierra, and M. Wooldridge. Automated negotiation: Prospects, methods, and challenges. *Journal of Group Decision and Negotiation*, 2(10):199–215, 2001.
9. I. Rahwan. *Interest-based Negotiation in Multi-Agent Systems*. PhD thesis, De-partment of Information Systems, University of Melbourne, Melbourne, Australia, 2004.
10. I. Rahwan, S. Ramchurn, N. Jennings, P. McBurney, S. Parsons, and L. Sonenberg. Argumentation based negotiation. *Knowledge Engineering Review*, 18(4):343–375, 2003.
11. J. Rouchier. Reimplementation of a multi-agent model aimed at sustaining ex-perimental economic research: the case of simulations with emerging speculation. *Journal of Artificial Societies and Social Simulation*, 6(4), 2003.
12. K. M. Sim. A market-driven model for designing negotiation agents. *Computa-tional Intelligence*, 4(18):618–637, 2002.
13. B. Vitteau and M.P. Huget. Modularity in interaction protocols. In F. Dignum, editor, *Advances in Agent Communication, International Workshop on Agent Com-munication Languages (AC 2003)*, volume 2922 of *Lecture Notes in Artificial In-telligence (LNAI)*, pages 291–309. Springer-Verlag, 2004.
14. P. Winoto, G. McCalla, and J. Vassileva. Non-monotonic-offers bargaining pro-tocol. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, pages 1072–1079, 2004.