Minimally intrusive negotiating agents for resource sharing*

Fariba Sadri **Francesca Toni**
Department of Computing
Imperial College
180, Queens Gate
SW7 London, UK
{fs, ft}@doc.ic.ac.uk

Paolo Torroni
DEIS
University of Bologna
V.le Risorgimento, 2
40136 Bologna, Italy
ptorroni@deis.unibo.it

Abstract

We study the problem of agents negotiating periods of time during which they can have use of resources, thus allowing for the sharing of resources. We define a multi-stage negotiation framework where agents, in order to obtain resources, step through a sequence of stages, each characterised by an increased chance of a mutually agreeable deal but at the price of disclosing more and more information. In the sequence, the agents may agree to move to the next stage if the previous stage fails to produce a deal amongst them. In this paper, we concentrate on two early negotiation stages, characterised by minimal disclosure of information. Thus, the agents negotiating at these stages can be thought of as "minimally intrusive".

1 Introduction

Negotiation of resources is an important research area in multi-agent systems. In general, agents may negotiate to obtain resources that they are missing but that are necessary to carry out their plans. In this paper, we assume that negotiation for resources takes place within the framework of [Sadri *et al.*, 2002], where the knowledge of the agents is represented as a tuple $< \mathcal{B}, \mathcal{R}, \mathcal{I}, \mathcal{D}, \mathcal{G} >$ with $\mathcal{B}$: *beliefs* about the world, the self and the other agents, as well as the negotiation policies of the agent; $\mathcal{R}$: initially owned *resources*; $\mathcal{I}$: *intentions*, i.e., the plans[1] that the agent intends to carry out, in order to achieve its goals, together with the resources required for that plan; $\mathcal{D}$: store of past *dialogues*; and $\mathcal{G}$: agent *goals*.

We extend [Sadri *et al.*, 2002] to negotiate not just resources, but also time windows during which resources can be used by agents, thus allowing for sharing of (non-consumable) resources over time, The extended framework allows solutions for a wider range of resource re-allocation problems. In the extended framework, agent intentions are sets of activities. Drawing inspiration from [El Sakkout and Wallace, 2000], we model an activity a (a is a unique identifier) as a tuple $< a, R_a, D_a, Ts_a, Te_a >$ denoting that a requires resource R_a, has duration D_a, earliest start time Ts_a, latest end time Te_a. Note that, without loss of generality, we assume that each activity requires only one resource [2]. In the extended framework, the knowledge of agents includes a (possibly empty) *concrete schedule* $\hat{\mathcal{I}}$ for activities, where if $< a, t_s, t_e > \in \hat{\mathcal{I}}$ then $< a, R, D, Ts_a, Te_a > \in \mathcal{I}$, such that $t_e - t_s = D$, $t_s \geq Ts_a$, and $t_e \leq Te_a$.

We develop negotiation protocols and policies that allow agents to exchange resources and strike deals for agreed time windows. We introduce a multi-stage process of negotiation, each characterised by a protocol and a policy. A higher stage involves more information passing between agents and more sophisticated negotiation, possibly including re-planning by the agents aimed at trying to help satisfy each other's constraints and requirements. The following example illustrates the sort of problems and solutions we propose in this paper.

Example 1 Let x and y be two agents. x's intentions contain an activity a requiring a resource r, $< a, r, 3, 1, 5 > \in \mathcal{I}_x$. y's intentions contain an activity b also requiring r. x owns r from time 1 to 3 and from time 5 onwards, and y owns r from time 3 to 5. x needs r for three consecutive time slots between 1 and 5, but it currently owns r for only two time slots. Let us consider the three scenarios below.

1. $< b, r, 1, 4, 10 > \in \mathcal{I}_y$. The problem is solved by y agreeing to give r to x for $[3, 4]$.
2. $< b, r, 1, 3, 10 > \in \mathcal{I}_y$ and y has a concrete schedule $\hat{\mathcal{I}}_y =< b, 3, 4 >$ for b to be carried out between 3 and 4. The problem is solved by y agreeing to postpone its schedule by 1 time slot and giving r to x for $[3, 4]$.
3. $< b, r, 2, 3, 10 > \in \mathcal{I}_y$. y cannot give r away, because otherwise b becomes unfeasible. The problem can be solved by an exchange: y agrees to give r away to x for $[3, 4]$, in return for x giving r to y for $[5, 6]$. □

2 Background and Preliminaries

In this section we review and adapt some concepts from background papers, needed in the rest of the paper.

*This work is partially funded by the IST programme of the European Commission under the IST-2001-32530 SOCS project.

[1]It is beyond the scope of this paper to provide a general and exhaustive representation of the agent knowledge. Instead, we concentrate on those elements relevant to the resource sharing problem. In particular, for simplicity, we identify here intentions with plans.

[2]Indeed, it is possible to model any activity requiring multiple resources by a number of activities, one for each resource, with the same duration and times.

Definition 1 An *agent system* is a finite set $\mathcal{S}$, with at least two elements, where each $x \in \mathcal{S}$ is a ground term, representing the name of an agent. All elements of $\mathcal{S}$ are distinct (agent names must be uniquely identifiable). Each agent $x \in \mathcal{S}$ is equipped at each time $\tau \in \mathbb{N}$ with a *knowledge base* $\mathcal{K}_{x,\tau}$, namely a tuple $< \mathcal{B}_x, \mathcal{R}_x, \mathcal{I}_x, \mathcal{D}_{x,\tau}, \mathcal{G}_x, \hat{\mathcal{I}}_{x,\tau} >$, as explained in the introduction.

Note that, in this paper, we assume that the only parts of the knowledge base of an agent that change over time are the dialogue store, which grows in time, and the concrete schedule. We will also assume that, in $\mathcal{I}_x$, all activities that require the same resource have disjoint time windows. In the sequel, when clear from the context, we will sometimes refer to the knowledge base of an agent x simply as $\mathcal{K}_x$ or $\mathcal{K}$.

Definition 2 Let $\mathcal{S}$ be an agent system. Let $\mathcal{R}es$ be a set of resources.[3] Let $\mathcal{R}es \times \mathbb{N} \times \mathbb{N} \longrightarrow \mathcal{S}$ be the resource allocation in the system at time τ (this can be determined, for example, from $\mathcal{R}$ and $\mathcal{D}$ of all agents at time τ). The allocation of resources is defined for time periods $[T_s, T_e]$, $T_s \geq 0, T_e \leq T_{max}$, where T_{max} is the maximum scheduling time, defined as follows:

$$T_{max} \stackrel{def}{=} \max_{x \in \mathcal{S}} \{TE \mid < \_, \_, \_, \_, TE > \in \mathcal{I}_x\}.$$

The negotiation process we will define modifies the resource distribution in the agent system through time. The *Temporal Resource Reallocation Problem (T-RRP)* is the problem of answering to the following *question*: Does there exist a time τ during the negotiation process when the resource distribution is such that each agent has the resources it requires for time periods that would allow it to perform the activities in its intention, within their specified time windows?

The purpose of this work is to show how it is possible to find a solution to the T-RRP (when one exists) by using agents that negotiate by means of dialogues. We will assume that all agents in a system share the same communication language, in terms of syntax, semantics, ontology, and pragmatics. We refer to [Sadri *et al.*, 2002] for a formal definition of a language for negotiation. In brief, a language defines the set of allowed dialogue moves. Each dialogue move is coded into a $tell$ predicate, which has 5 arguments: sender, receiver, content, dialogue identifier, and time of the move. We will use the communication language $\mathcal{L}_{1-TW}$, defined below:

Definition 3 $\mathcal{L}_{1-TW} = \{$
$tell(X, Y, Content, D, T) \mid Content \in$
$\{$ (1) $\mathbf{request}(\mathbf{give}(R, (Ts, Te)))$
(2) $\mathbf{accept}(\mathbf{request}(\mathbf{give}(R, (Ts, Te))))$
(3) $\mathbf{refuse}(\mathbf{request}(\mathbf{give}(R, (Ts, Te))))$
(4) $\mathbf{promise}(R, (Ts, Te), (Ts', Te'))$
(5) $\mathbf{change}(\mathbf{promise}(R, (Ts, Te), (Ts', Te')))$
(6) $\mathbf{accept}(\mathbf{promise}(R, (Ts, Te), (Ts', Te')))$ $\}\}$
where Ts, Te are positive integers between 0 and T_{max}, for all dialogue moves, $Ts < Te$, and for (4)–(6), $Ts' < Te'$.

The first three moves are of intuitive meaning. (4) is used by X to propose a deal (*promise*): X will give R to Y for the interval $[Ts', Te']$ if Y will give R to X for the interval $[Ts, Te]$. (5) is used to refuse a proposed deal *and* ask for a new one (there is no *refuse_promise* move which only terminates a protocol), (6) is used to accept a deal.

[3]Without loss of generality, we assume that all resources in the system are non-consumable (for resource reallocation, consumable resources differ from non-consumable ones in that the former can be allocated only for maximal intervals $[0, T_{max}]$, and not for any sub-intervals, where T_{max} is given below in the definition).

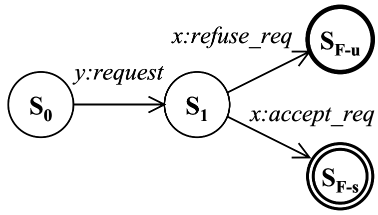

Figure 1: Protocol for Stage 1

Given a language for negotiation $\mathcal{L}$, we define the set of *final moves* $\mathcal{F}(\mathcal{L})$. In particular, $\mathcal{F}(\mathcal{L}_{1-TW})$ is the subset of $\mathcal{L}_{1-TW}$ that contains all the moves whose content is (2), (3) or (6).

Dialogues can be generated by means of policies, held by the knowledge base of the agents.

Definition 4 *Policies* are expressed as *dialogue constraints* of the form $p_i \wedge C \Rightarrow p_{i+1}$, where p_i and p_{i+1} are moves. The conditions C are to be evaluated in the knowledge base of the agent or in extensions to it. The intended use of these policies is that if the agent receives a move p_i, and the conditions C are satisfied in its knowledge base, the agent generates p_{i+1}. An operational model for policies is defined in [Sadri *et al.*, 2002].

Definition 5 Given an agent system $\mathcal{S}$ equipped with a language for negotiation $\mathcal{L}$ and two agents X and Y in $\mathcal{S}$ equipped with policy Pol, a *dialogue induced by* Pol between X and Y is a set of ground dialogue moves in $\mathcal{L}$, $\{p_0, p_1, p_2, \ldots\}$, such that, for a given set of time lapses $0 \leq \tau_0 < \tau_1 < \tau_2 < \ldots$:

1. $\forall\ i \geq 0$, p_i is uttered at time τ_i;
2. $\forall\ i \geq 0$, if p_i is uttered by agent X (viz. Y), then p_{i+1} (if any) is uttered by agent Y (viz. X);
3. $\forall\ i > 0$, p_i can be uttered by agent $U \in \{X, Y\}$ only if there exists a (grounded) dialogue constraint in Pol, $p_{i-1} \wedge C \Rightarrow p_i$ s.t. $\mathcal{K}_{U,\tau_{i-1}} \wedge p_{i-1} \vdash C$;
4. there is an identifier D such that, $\forall\ i \geq 0$, the dialogue identifier of p_i is D;
5. $\forall \tau$, $\tau_{i-1} < \tau < \tau_i, \forall i > 0$ s.t. p_i and p_{i-1} belong to the dialogue, there exist no moves at τ with either X or Y being either the receiver or the utterer.

A dialogue $\{p_0, p_1, \ldots p_m\}$, $m \geq 0$, is *terminated* if p_m is a ground final move, namely p_m is a ground instance of a move in $\mathcal{F}(\mathcal{L})$. We say that such a terminated dialogue *starts* at time τ_0 and *ends* at time τ_m.

Note that this definition prevents agents from being involved in more than one dialogue at a time (e.g., dialogues cannot be nested).

An important property of policies used to induce dialogues as in Definition 5 is *conformance* to protocols, known to all agents involved in the dialogue. A dialogue protocol can be defined as a set of states, representing the current state of dialogue, a set of allowed dialogue moves, and a set of transition rules that, given a state and a move, produce a state. A protocol is therefore defined as a finite state machine, consisting of states and arcs, which has among its states an initial state

S_0, two final states, S_{F-s} (successful termination) and S_{F-u} (unsuccessful termination), and possibly a number of intermediate states S_i. The arcs can be viewed as allowed transitions mapping one state to another given a label. These labels correspond to the content of moves. An example of protocol is in Fig. 1. When we show the protocols, we use some abbreviations, such as for instance:

request for *request*(*give*$(R, (Ts, Te))$),
refuse_req for *refuse*(*request*(*give*$(R, (Ts, Te))$)).

In order to define the concept of conformance, we define dialogues in relation with protocols.

Definition 6 Given an agent system $\mathcal{S}$ equipped with a language for negotiation $\mathcal{L}$, and a protocol $\mathcal{P}$, a *dialogue conforming to* $\mathcal{P}$, between two agents X and Y in $\mathcal{S}$, is a set of ground dialogue moves in $\mathcal{L}$, $\{p_0, p_1, p_2, \ldots\}$, such that, for a given set of time lapses $0 \leq \tau_0 < \tau_1 < \tau_2 < \ldots$:

1. 2. 4. 5. as for Definition 5;
3. the content of p_0 must label an arc from S_0. $\forall i$, if the content of p_i is the label of an arc into a final state, then there is no p_{i+1} in the dialogue. $\forall$ i, the contents of p_i and p_{i+1}, if they both exist, must be labels, respectively, of an arc going into a state, and an arc coming out of the same state.

A dialogue $\{p_0, p_1, \ldots p_m\}$, $m \geq 0$, is *terminated* if p_m is the label of an arc into a final state.

We are now ready to define the concept of conformance of policies to protocols.

Definition 7 Given a policy Pol and a protocol $\mathcal{P}$, Pol *conforms* to $\mathcal{P}$ if every dialogue induced by Pol is a dialogue conforming to $\mathcal{P}$.

We define sequences of dialogues for a resource R and an activity A, between two agents X and Y.

Definition 8 Given an agent system $\mathcal{S}$, a *sequence of dialogues* σ between two agents X and Y in $\mathcal{S}$ for a resource R and an activity A is a set of terminated dialogues between X and Y, $\sigma = \{d_0, d_1, d_2, \ldots\}$, where $d_j = \{p_{j0}, p_{j1}, \ldots, p_{jn}\}$ for all j, such that, for a given set of time lapses $0 \leq \tau_0 < \tau_1 < \tau_2 < \ldots$:

1. $\forall$ $i \geq 0$, d_i is initiated by X at time τ_i, and it is terminated at a time $\tau'_i < \tau_{i+1}$;
2. $\nexists i, j, i \neq j$ such that $p_{i0} = p_{j0}$;
3. $\forall i$, if d_i and d_{i+1} are in σ, then $d_i = \{p_{i0}, \ldots, p_{in}\}$ and $p_{in} =$ *tell*$(Y, X,$ *refuse*(*request*(*give*$(R, (Ts, Te)$))), $d_i, \tau)$ for some Ts, Te, τ;
4. $\exists < A, R, D, TS, TE > \in \mathcal{I}_X$ such that $\forall i$ $p_{i0} =$ *tell*$(X, Y,$ *request*(*give*$(R, (Ts, Te)$)), $d_i, \tau_i)$) for some Ts, Te such that $Ts \geq TS$ and $Te \leq TE$;
5. If σ is finite, i.e. $\sigma = d_0, \ldots, d_f$ for some $f \geq 0$, then $d_f = \{p_{f0}, \ldots, p_{fn}\}$, the last dialogue in the sequence, is such that for some $\tau, T_s, T_e, T'_s, T'_e$, $p_{fn} =$ *tell*$(Y, X,$ *accept*(*request*(*give*$(R,$ $(Ts,$ $Te)$))), d_f, $\tau)$ or $p_{fn} =$ *tell*$(Y, X,$ *accept*(*promise*$(R,$ (Ts, Te), (Ts', Te'))), $d_f, \tau)$.

We assume the atomicity of sequences of dialogues: agents will not react to any incoming request about a resource r if they are participating in an ongoing sequence of dialogues regarding r, and moreover they themselves will not make a request for r with respect to another activity while they are participating in an ongoing sequence of dialogues which they initiated (an agent cycle very similar to the one in [Sadri *et al.*, 2002] will achieve this atomicity).

3 Negotiation Stages

We define two different stages of negotiation, each characterized by the degree of flexibility of the agents and the amount of information disclosed and used by them:

Stage 1: Request/flexible schedule
Stage 2: Blind deal

In this section, for each stage we define the *protocol*, the *policies* adopted by the agents, and the *properties* of the stage. The properties that we study are (i) conformance of the policy to the protocol, (ii) properties of single dialogues (termination and characterization of the class of problems that can be solved), (iii) properties of sequences of dialogues happening at that stage, and (iv) subsumption of earlier stages (in terms of solvable problems). In particular, for each stage we give an example of a problem that can be solved within it, and an example of a problem that cannot.

In defining the policies and in stating the results, we rely upon some predicates, whose formal definition is given in appendix. We use the notation $\mathcal{K}_{X,\tau} \vdash p$ to indicate that at time τ the knowledge of X entails a certain predicate p. The proofs of the results are omitted for lack of space.

3.1 Stage 1 - request/flexible schedule

Protocol. The protocol is given in Figure 1.

Policy. The policy is shown in Figure 2: an agent will *accept* a *request* for R if it can find a concrete schedule of its own activities that does not make use of R during the requested interval. It will *refuse* it otherwise, leading to an unsuccessful final state. Note that it is up to the requesting agent to find good heuristics to formulate a request or a series of requests which can be accepted.

Example 2 Let us consider the following example:

$\mathcal{I}_y = \{< a, r, 5, 10, 20 >\}$ $\quad$ $\mathcal{I}_x = \{< b, r, 5, 10, 20 >\}$
$\hat{\mathcal{I}}_y = \{< a, 11, 16 >\}$ $\quad$ $\hat{\mathcal{I}}_x = \emptyset$
$\mathcal{R}_y = \{have(r, (10, 20))\}$ $\quad$ $\mathcal{R}_x = \emptyset$

There is no solution if y sticks to $\hat{\mathcal{I}}_y$: but if y was happy with a different schedule, e.g. $\hat{\mathcal{I}}'_y = \{< a, 15, 20 >\}$, then x and y could both do their activities by sharing resource r. Thus, this example can be solved by the following negotiation dialogue $d(1)$ occurring at Stage 1 (i.e., induced by the policy of Fig. 2):

tell$(x, y,$ *request*(*give*$(r, (10, 15)))$, $d(1), 1)$
tell$(y, x,$ *accept*(*request*(*give*$(r, (10, 15))$))), $d(1), 2)$ □

Properties. Stage 1 is computationally demanding for the agent who is replying to a request. The problems that can be solved at this stage are all those that can be solved by means of a (possibly empty) modification in the agents' current concrete schedule.

(i)*: protocol conformance.* It is possible to prove that the policy of Stage 1 is conforming to the protocol in Fig. 1.

(ii)*: properties of single dialogues.* A request/flexible schedule interaction enjoys the property of termination. In fact, dialogues following this protocol have a fixed number of steps.

Theorem 1 Let us consider a system composed of two agents, x and y. Then, for all system resources r, all activities

$(IC.1)$ $\textit{tell}(X, y, \mathbf{request}(\mathbf{give}(R, (Ts, Te))), D, T) \wedge \textit{have}(R, (Ts, Te), T) \wedge \neg\, \textit{need}(R, (Ts, Te), T)$
$\quad \Rightarrow\ \exists\, T' \mid \textit{tell}(y, X, \mathbf{accept}(\mathbf{request}(\mathbf{give}(R, (Ts, Te)))), D, T') \wedge T < T'$

$(IC.2)$ $\textit{tell}(X, y, \mathbf{request}(\mathbf{give}(R, (Ts, Te))), D, T) \wedge \textit{have}(R, (Ts, Te), T) \wedge \textit{need}(R, (Ts, Te), T)$
$\quad \Rightarrow \exists\, T' \mid \textit{tell}(y, X, \mathbf{refuse}(\mathbf{request}(\mathbf{give}(R, (Ts, Te)))), D, T') \wedge T < T'$

$(IC.3)$ $\textit{tell}(X, y, \mathbf{request}(\mathbf{give}(R, (Ts, Te))), D, T) \wedge \neg\, \textit{have}(R, (Ts, Te), T)$
$\quad \Rightarrow \exists\, T' \mid \textit{tell}(y, X, \mathbf{refuse}(\mathbf{request}(\mathbf{give}(R, (Ts, Te)))), D, T') \wedge T < T'$

Figure 2: *Stage 1* policy for an agent y

a assigned to x, all times τ, and all intervals $[Ts, Te]$ s.t.
$\mathcal{K}_{x,\tau} \vdash \textit{miss}(r, (Ts, Te), a) \;\wedge$
$\mathcal{K}_{y,\tau} \vdash \textit{avail}(r, (Ts, Te))$,
there exists a dialogue d induced by the policy of Stage 1, starting at time τ, and ending at time τ', such that
$\mathcal{K}_{x,\tau'} \vdash \textit{need}(r, (Ts, Te), a, \tau') \;\wedge$
$\mathcal{K}_{y,\tau'} \vdash \textit{indiff}(r, (Ts, Te))$. Intuitively this theorem states that if there exists a time window tw such that an agent x needs r in it and another agent has r available in tw, then there also exists a dialogue induced by the policy of Stage 1 which solves x's reallocation problem about r. This intuitive understanding of the theorem is the result of the formal predicate definitions given in appendix.

(iii)*: properties of sequences.* It is up to the agent who is missing a resource to find good heuristics to formulate the requests, which can lead to a successful dialogue sequence (provided it is possible to find a solution at this stage). Given an agent x, a resource r, and an activity a, one possibility could be a cycle starting at a time τ where x successively asks for intervals $[Ts, Te]$ such that $\mathcal{K}_{x,\tau} \vdash \textit{miss}(r, (Ts, Te), a)$, until all such intervals are exhausted, or the request for one of the intervals is accepted. We call this *Strategy1*.

Theorem 2 Let us consider a system composed of two agents, x and y, negotiating at Stage 1. Then, for all resources r in the system, and all activities a assigned to x that require r, if x follows *Strategy1* in its attempt to acquire r from y, starting at τ with a request for r for an interval $[Ts, Te]$ such that $\mathcal{K}_{x,\tau} \vdash \textit{miss}(r, (Ts, Te), a)$, then either the resulting sequence of dialogues will be finite and will terminate at a time τ', and $\mathcal{K}_{x,\tau'} \vdash \textit{feas}(a, (Ts^{\star}, Te^{\star}))$, for some $Ts^{\star}, Te^{\star}$, or $\nexists\, Ts', Te'$ such that
$\mathcal{K}_{x,\tau} \vdash \textit{miss}(r, (Ts', Te'), a) \wedge$
$\mathcal{K}_{y,\tau} \vdash \textit{avail}(r, (Ts', Te'))$.

(iv)*: subsumption of earlier stages.* Stage 1 subsumes our previous work done in [Sadri *et al.*, 2002], in the sense that the resource reallocation problems it solves include those solved by [Sadri *et al.*, 2002].

We now give a counterexample for Stage 1.

Example 3 Let us consider the following modification of Example 2, where there is a different initial resource assignment, and different time windows for the activity b (b must be completed by 15):

$$\mathcal{I}_y = \{< a, r, 5, 10, 20 >\} \qquad \mathcal{I}_x = \{< b, r, 5, 10, 15 >\}$$
$$\mathcal{R}_y = \{have(r, (10, 15))\} \qquad \mathcal{R}_x = \{have(r, (15, 20))\}$$

We do not give any concrete schedules for the agents' activities since they play no role here. There is no solution to this problem that can be found at Stage 1. In fact, if y gives away r, it will not be possible for it to carry out a any more. □

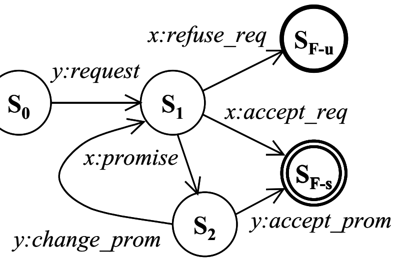

Figure 3: Protocol for Stage 2

3.2 Stage 2 - blind deal

Protocol. The protocol is shown in Fig. 3. After the initial move by y, *request*(*give*$(R, (Ts, Te))$), which makes the dialogue reach state S_1, the other agent x can either *accept* or *refuse*, as in Stage 1, or can propose a deal (*promise*). After a promise, the agent y who made the request can either accept the deal, causing a successful termination, or refuse it (*change_prom*), which brings back to S_1.

Policy. The policy is shown in Fig. 4.[4] An agent will *accept* a request about a resource R for the period $[Ts, Te]$ if it would do it at Stage 1, but it will refuse it *only if* it does not have any deal to propose (*promise*). In particular, an agent x will propose a deal, in reply to a request made by an agent y, if there exists an interval $[Ts', Te']$, disjoint from $[Ts, Te]$, which has the following property: Once y obtains R for $[Ts', Te']$, it will not need it anymore for $[Ts, Te]$. In that case, if y accepts to give R away for the interval $[Ts', Te']$, the negotiation process reaches a successful final state, otherwise x may continue proposing different deals (if they exist), until y accepts one (successful termination), or there exist no new ones to propose, in which case x will refuse the initial request, thus leading to an unsuccessful final state.

Since there might be several alternative proposals for a deal at a given time, but we want the agent's policies to be deterministic, we use in the definition of the policies of an agent x at Stage 2 a predicate *pick*$((Ts, Te), T)$, that at any given time T uniquely determines a time period, having made reference possibly to $\mathcal{D}_{x,T}$ and $\mathcal{R}_x$.

Example 3 can be solved by the following negotiation dialogue $d(2)$ occurring at Stage 2:
tell$(x, y,$ *request*$($*give*$(r, (10, 15))), d(2), 1)$
tell$(y, x,$ *promise*$(r, (15, 20), (10, 15)), d(2), 2)$
tell$(x, y,$ *accept*$($*promise*$(r, (15, 20), (10, 15))), d(2), 3)$

Properties. At Stage 2 agents are more cooperative than at Stage 1. This is achieved by both agents – and in particu-

[4] We use the notation $\mathcal{K}_{X,\tau} \cup \Delta \vdash p$, where Δ is a set of atoms, to mean that $\mathcal{K}_{X,\tau}$, enlarged with Δ, entails p.

$(IC.1)$ & $(IC.3)$ from Stage 1

$(IC.4)$ $tell(X, y, \mathbf{request}(\mathbf{give}(R, Ts, Te)), D, T) \wedge have(R, (Ts, Te), T) \wedge need(R, (Ts, Te), A, T)$
$\wedge \nexists Ts', Te' \mid \neg have(R, (Ts', Te')) \wedge tell(y, X, promise(R, (Ts', Te'), (Ts, Te)), D, \_) \notin \mathcal{D}_{y,T}$
$\wedge \mathcal{K}_{y,T} \cup \{obtain(R, (Ts', Te')), give\_away(R, (Ts, Te))\} \vdash feas(A, (Ts', Te'))$
$\Rightarrow \exists T' \mid tell(y, X, \mathbf{refuse}(\mathbf{request}(\mathbf{give}(R, Ts, Te))), D, T') \wedge T < T'$

$(IC.5)$ $tell(X, y, \mathbf{request}(\mathbf{give}(R, Ts, Te)), D, T) \wedge have(R, (Ts, Te), T) \wedge need(R, (Ts, Te), A, T)$
$\wedge pick((Ts', Te'), T) \wedge \neg have(R, (Ts', Te'), T) \wedge tell(y, X, promise(R, (Ts', Te'), (Ts, Te)), D, \_) \notin \mathcal{D}_{y,T}$
$\wedge \mathcal{K}_{y,T} \cup \{obtain(R, (Ts', Te')), give\_away(R, (Ts, Te))\} \vdash feas(A, (Ts', Te'))$
$\Rightarrow \exists T' \mid tell(y, X, \mathbf{promise}(R, (Ts', Te'), (Ts, Te)), D, T') \wedge T < T'$

$(IC.6)$ $tell(X, y, \mathbf{change}(\mathbf{promise}((\_, \_), (Ts, Te))), D, T) \wedge have(R, (Ts, Te), T) \wedge need(R, (Ts, Te), A, T)$
$\wedge pick((Ts', Te'), T) \wedge \neg have(R, (Ts', Te'), T) \wedge tell(y, X, promise(R, (Ts', Te'), (Ts, Te)), D, \_) \notin \mathcal{D}_{y,T}$
$\wedge \mathcal{K}_{y,T} \cup \{obtain(R, (Ts', Te')), give\_away(R, (Ts, Te))\} \vdash feas(A, (Ts', Te'))$
$\Rightarrow \exists T' \mid tell(y, X, \mathbf{promise}(R, (Ts', Te'), (Ts, Te)), D, T') \wedge T < T'$

$(IC.7)$ $tell(X, y, \mathbf{change}(\mathbf{promise}(R, (Ts', Te'), (Ts, Te))), D, T) \wedge have(R, (Ts, Te), T) \wedge need(R, (Ts, Te), A, T)$
$\wedge \nexists Ts', Te' \mid \neg have(R, (Ts', Te'), T) \wedge tell(y, X, promise(R, (Ts', Te'), (Ts, Te)), D, \_) \notin \mathcal{D}_{y,T}$
$\wedge \mathcal{K}_{y,T} \cup \{obtain(R, (Ts', Te')), give\_away(R, (Ts, Te))\} \vdash feas(A, (Ts', Te'))$
$\Rightarrow \exists T' \mid tell(y, X, \mathbf{refuse}(\mathbf{request}(\mathbf{give}(R, (Ts, Te)))), D, T') \wedge T < T'$

$(IC.8)$ $tell(X, y, \mathbf{promise}(R, (Ts', Te'), (Ts, Te)), D, T) \wedge miss(R, (Ts, Te), A)$
$\wedge \mathcal{K}_{y,T} \cup \{obtain(R, (Ts, Te)), give\_away(R, (Ts', Te'))\} \vdash feas(A, (\_, \_))$
$\Rightarrow \exists T' \mid tell(y, X, \mathbf{accept}(\mathbf{promise}(R, (Ts', Te'), (Ts, Te)), D, T')) \wedge T < T'$

$(IC.9)$ $tell(X, y, \mathbf{promise}(R, (Ts', Te'), (Ts, Te)), D, T) \wedge miss(R, (Ts, Te), A)$
$\wedge \mathcal{K}_{y,T} \cup \{obtain(R, (Ts, Te)), give\_away(R, (Ts', Te'))\} \nvdash feas(A, (\_, \_))$
$\Rightarrow \exists T' \mid tell(y, X, \mathbf{change}(\mathbf{promise}(R, (Ts', Te'), (Ts, Te)), D, T')) \wedge T < T'$

Figure 4: *Stage 2* policy for an agent y

lar the agent to whom the request is addressed – to do more reasoning in order to be helpful in response to requests.

(i)*: protocol conformance.* It is possible to prove that the policy of Stage 2 is conforming to the protocol in Fig. 3.

(ii)*: properties of single dialogues.* Stage 2 does not enjoy the fixed dialogue length property of Stage 1, but it terminates if we have a finite scheduling horizon T_{max} and we do not allow the same move twice at different times, in the same dialogue (which is a reasonable requirement). Stage 2 is computationally more demanding than Stage 1 for both agents.

Theorem 3 Let us consider a system composed of two agents, x and y, each having an initial resource assignment. Then, for all resources r in the system, all activities a and b, all times τ and all intervals $[Ts, Te]$, such that

$\mathcal{K}_{x,\tau} \vdash miss(r, (Ts, Te), a) \wedge$
$\mathcal{K}_{y,\tau} \vdash need(r, (Ts, Te), b) \wedge$
$\exists [Ts', Te']$ such that $\mathcal{K}_{y,\tau} \cup \{give\_away(r, (Ts, Te)), obtain(r, (Ts', Te'))\} \vdash feas(b, (Ts'', Te''))$

for some $[Ts'', Te'']$, there exists a dialogue d induced by the policy of Stage 2, starting at time τ, and ending at a time τ', such that

$\mathcal{K}_{x,\tau'} \vdash need(r, (Ts, Te), a) \wedge$
$\mathcal{K}_{y,\tau'} \vdash indiff(r, (Ts, Te))$.

(iii)*: properties of sequences.*

Theorem 4 Let us consider a system composed of two agents, x and y, negotiating at Stage 2. Then, for all resources r in the system, and all activities a assigned to x which require r, if starting at time τ x follows *Strategy1* to request r from y, either the sequence of dialogues will terminate at τ' after a finite number of dialogues, and $\mathcal{K}_{x,\tau'} \vdash feas(a, (Ts^\star, Te^\star))$ for some $Ts^\star, Te^\star$, or $\nexists Ts, Te, Ts', Te'$ such that

$\mathcal{K}_{x,\tau} \vdash miss(r, (Ts, Te), a) \wedge$
$\mathcal{K}_{y,\tau} \cup \{give\_away(r, (Ts, Te)), obtain(r, (Ts', Te'))\} \vdash indiff(r, (Ts, Te))$.

(iv)*: subsumption of earlier stages.* More problems can be solved by Stage 2 than by Stage 1, namely:

Theorem 5 Let us consider a system composed of two agents, x and y. For all resources r in the system, all activities a assigned to x, all times τ and intervals $[Ts, Te]$ s.t.

$\mathcal{K}_{x,\tau} \vdash miss(r, (Ts, Te), a) \wedge$
$\mathcal{K}_{y,\tau} \vdash avail(r, (Ts, Te))$,

there exists a dialogue d induced by the policy of Stage 2, starting at time τ, and ending at time τ', such that

$\mathcal{K}_{x,\tau'} \vdash need(r, (Ts, Te), a) \wedge$
$\mathcal{K}_{y,\tau'} \vdash indiff(r, (Ts, Te))$

Although this enlarges the set of problems that can be solved, it does not solve, for instance, the problems where more than one exchange is needed. We now give the following counterexample for Stage 2.

Example 4 Let us consider the following modification of Example 2, where there is a different resource and activity assignment:

$$\mathcal{I}_y = \{< a, r, 2, 10, 55 >, < c, r, 2, 15, 20 >, \} \quad \mathcal{I}_x = \{< b, r, 5, 10, 20 >\}$$
$$\mathcal{R}_y = \{have(r, (13, 17))\} \quad \mathcal{R}_x = \{have(r, (10, 13)), have(r, (17, 20))\}$$

There is no solution to this problem that can be found at Stage 2. In fact, the (minimal) requests that x will make to obtain r for 5 consecutive time periods are: $[13, 15]$ and $[15, 17]$. y may reply to the first one with a deal to obtain r for $[10, 12]$

or for $[11, 13]$, neither of which makes x's activity b feasible. The same will happen for the second request, which results in an unsuccessful sequence of negotiation dialogues for a problem that has a solution. □

4 Discussion

This work benefits from a logic-based high level approach that facilitates specification and proof of formal properties and which has an operational model, that forms a bridge between the system description and implementation. This feature is difficult to find in most related work: as the negotiation process becomes more elaborated, it gets harder to find and prove formal properties as those identified here. In the following, we briefly survey similar approaches to agent negotiation or proposed solutions to the same problem.

Our modelling of constraints draws inspiration from work on constraint satisfaction for a monolithic system [El Sakkout and Wallace, 2000]. But our work differs from it both in context and approach. Our aim is to provide protocols and policies for a multi-stage process of negotiation in a collaborative multi-agent context, whereas their aim is the solution of T-RRP problems viewed as constraint satisfaction problems, while minimising changes to concrete schedules. [Harrison, 2000] extends such work to multi-agent systems, but the focus there is again that of minimising changes to existing schedules, and not on negotiation.

[Conry *et al.*, 1992] proposes an approach to negotiation based on multiple stages, focusing on coordination degrees. Progressive stages require that agents solve problems in a more coordinated way. In the first stage, the agents try to solve problems independently of other agents' constraints, while in the last one it is possible, e.g., to discover that the overall problem is over-constrained and thus a certain goal is unfeasible.

There are many issues regarding negotiation that our paper does not address. For example, we do not deal with the efficiency and timeliness of negotiation. These issues are addressed in [Kraus *et al.*, 1995] using utility functions.

We are currently working on extending our negotiation process to a number of further stages, whereby the agents disclose to each other more information about their constraints in order to be able to propose more informed deals for exchanges of resources. We would like to stress that we approach a resource sharing problem from a multi-agent perspective: Existing scheduling techniques are likely to outperform compared to the negotiation processes outlined in this work, but they do not generally allow for agent autonomy. In the final stage of such an approach, where all the constraints are known, the problem becomes one of distributed constraint satisfaction. [Yokoo and Hirayama, 2000] reviews a number of algorithms for such an application.

References

[Conry *et al.*, 1992] S.E. Conry, K. Kuwabara, V.R. Lesser, and R.A. Meyer. Multistage negotiation in distributed constraint satisfaction. *IEEE TSMC*, 21(6), 1992.

[El Sakkout and Wallace, 2000] H.H. El Sakkout and M.G. Wallace. Probe backtrack search for minimal perturbation in dynamic scheduling. *Constraints*, 5(4), 2000.

[Harrison, 2000] C.A. Harrison. *Distributed Optimisation in a Brokered Architecture*. PhD diss., IC-PARC, Imperial College London, 2000.

[Kraus *et al.*, 1995] S. Kraus, J. Wilkenfeld, and G. Zlotkin. Multiagent negotiation under time constraints. *AIJ*, 75(2), 1995.

[Sadri *et al.*, 2002] F. Sadri, F. Toni, and P. Torroni. An abductive logic programming architecture for negotiating agents. In *Proc. JELIA*, LNCS 2424. Springer Verlag, 2002.

[Yokoo and Hirayama, 2000] M. Yokoo and K. Hirayama. Algorithms for distributed constraint satisfaction: A review. *AAMASJ*, 3(2), 2000.

Appendix A

$\mathcal{K}_{X,\tau} \vdash \mathbf{have}(\mathbf{R}, (\mathbf{Ts}, \mathbf{Te}), \tau) \leftrightarrow$
$\quad (Te \textit{ is } Ts + 1 \wedge \mathcal{K}_{X,\tau} \vdash \textit{have}(R, Ts)) \vee$
$\quad (Tn \textit{ is } Ts + 1 \wedge Tn < Te \wedge \mathcal{K}_{X,\tau} \vdash \textit{have}(R, (Tn, Te), \tau) \wedge$
$\qquad \mathcal{K}_{X,\tau} \vdash \textit{have}(R, Ts))$

$\mathcal{K}_{X,\tau} \vdash \mathbf{have}(\mathbf{R}, \mathbf{T}) \leftrightarrow$
$\quad (\mathcal{K}_{X,\tau} \vdash \textit{have\_initially}(R, T) \wedge$
$\qquad \nexists\, \tau' \leq \tau \mid \mathcal{K}_{X,\tau'} \vdash \textit{give\_away}(R, (Ta_{giv}, Tb_{giv}))) \wedge$
$\qquad Ta_{giv} \leq T \wedge Tb_{giv} > T) \vee$
$\quad (\exists\, \tau' \leq \tau \mid \mathcal{K}_{X,\tau'} \vdash \textit{obtain}(R, (Ta_{obt}, Tb_{obt}))) \wedge$
$\qquad Ta_{obt} \leq T \wedge Tb_{obt} > T \wedge \nexists\, \tau'' \leq \tau \mid$
$\qquad \mathcal{K}_{X,\tau''} \vdash \textit{give\_away}(R, (Ta_{giv}, Tb_{giv}))) \wedge$
$\qquad Ta_{giv} \leq T \wedge Tb_{giv} > T \wedge \tau' \leq \tau'')$

$\mathcal{K}_{X,\tau} \vdash \mathbf{have\_initially}(\mathbf{R}, \mathbf{T}) \leftrightarrow$
$\quad \exists\, Ts, Te \mid \textit{have}(R, (Ts, Te)) \in \mathcal{R}_X \wedge Ts \leq T \wedge Te > T$

$\mathcal{K}_{X,\tau} \vdash \mathbf{give\_away}(R, (Ta_{giv}, Tb_{giv}))) \leftrightarrow$
$\quad \exists\, \textit{tell}(X, \_, Subject, \_, \tau) \in \mathcal{D}_{X,\tau} \mid$
$\quad Subject \in \{\textit{accept}(\textit{request}(\textit{give}(R, (Ta_{obt}, Tb_{obt})))),$
$\quad \textit{accept}(\textit{promise}(R, (Ta_{obt}, Tb_{obt}), (\_, \_)))\} \vee$
$\quad \exists\, \textit{tell}(\_, X, Subject, \_, \tau) \in \mathcal{D}_{X,\tau} \mid$
$\quad Subject \in \{\textit{accept}(\textit{promise}(R, (\_, \_), (Ta_{obt}, Tb_{obt})))\}$

$\mathcal{K}_{X,\tau} \vdash \mathbf{obtain}(R, (Ta_{obt}, Tb_{obt}))) \leftrightarrow$
$\quad \exists\, \textit{tell}(\_, X, Subject, \_, \tau) \in \mathcal{D}_{X,\tau} \mid$
$\quad Subject \in \{\textit{accept}(\textit{request}(\textit{give}(R, (Ta_{obt}, Tb_{obt})))),$
$\quad \textit{accept}(\textit{promise}(R, (Ta_{obt}, Tb_{obt}), (\_, \_)))\} \vee$
$\quad \exists\, \textit{tell}(X, \_, Subject, \_, \tau) \in \mathcal{D}_{X,\tau} \mid$
$\quad Subject \in \{\textit{accept}(\textit{promise}(R, (\_, \_), (Ta_{obt}, Tb_{obt})))\}$

$\mathcal{K}_{X,\tau} \vdash \mathbf{feas}(A, (Ts, Te)) \leftrightarrow$
$\quad \exists < A, R, D, Ts', Te' > \in \mathcal{I}_X \mid Ts' \leq Ts \wedge Te' \geq Te \wedge$
$\qquad Te - Ts = D \wedge \mathcal{K}_{X,\tau} \vdash \textit{have}(R, (Ts, Te), \tau)$

$\mathcal{K}_{X,\tau} \vdash \mathbf{need}(R, (Ts, Te), \tau) \leftrightarrow$
$\quad \exists < A, R, \_, \_, \_ > \in \mathcal{I}_X \mid \mathcal{K}_{X,\tau} \vdash \textit{need}(R, (Ts, Te), A)$

$\mathcal{K}_{X,\tau} \vdash \mathbf{need}(R, (Ts, Te), A) \leftrightarrow$
$\quad \mathcal{K}_{X,\tau} \vdash \textit{have}(R, (Ts, Te), \tau) \wedge$
$\quad \exists < A, R, D, Ts', Te' > \in \mathcal{I}_X \wedge Ts \geq Ts' \wedge Te \leq Te' \wedge$
$\quad \nexists\, Ts'' \in [Ts', Te' - D] \mid$
$\qquad \mathcal{K}_{X,\tau} \vdash \textit{have}(R, (Ts'', Ts'' + D), \tau) \wedge$
$\qquad (Ts'' \geq Te \vee Ts'' + D \leq Ts)$

$\mathcal{K}_{X,\tau} \vdash \mathbf{miss}(R, (Ts, Te)) \leftrightarrow$
$\quad \exists < A, R, \_, \_, \_ > \in \mathcal{I}_X \mid \mathcal{K}_{X,\tau} \vdash \textit{miss}(R, (Ts, Te), A)$

$\mathcal{K}_{X,\tau} \vdash \mathbf{miss}(R, (Ts, Te), A) \leftrightarrow$
$\quad \exists < A, R, \_, \_, \_ > \in \mathcal{I}_X \mid \mathcal{K}_{X,\tau} \nvdash \textit{feas}(A, (\_, \_)) \wedge$
$\qquad \mathcal{K}_{X,\tau} \cup \{\textit{obtain}(R, (Ts, Te))\} \vdash \textit{feas}(A, (\_, \_)) \wedge$
$\qquad \nexists\, (Ts', Te') \subset (Ts, Te) \mid$
$\qquad \mathcal{K}_{X,\tau} \cup \{\textit{obtain}(R, (Ts', Te'))\} \vdash \textit{feas}(A, (\_, \_))$

$\mathcal{K}_{X,\tau} \vdash \mathbf{indiff}(R, (Ts, Te)) \leftrightarrow$
$\quad \mathcal{K}_{X,\tau} \nvdash \textit{have}(R, (Ts, Te), \tau) \wedge$
$\quad \forall (Ts_j, Te_j) \mid \mathcal{K}_{X,\tau} \vdash \textit{miss}(R, (Ts_j, Te_j)) \rightarrow$
$\qquad Te \leq Ts_j \vee Ts \geq Te_j$

$\mathcal{K}_{X,\tau} \vdash \mathbf{avail}(R, (Ts, Te)) \leftrightarrow$
$\quad \mathcal{K}_{X,\tau} \vdash \textit{have}(R, (Ts, Te), \tau) \wedge \mathcal{K}_{X,\tau} \nvdash \textit{need}(R, (Ts, Te), \tau)$