Università degli Studi di Bologna
DEIS

A multi-stage negotiation architecture for sharing resources amongst logic-based agents

Fariba Sadri Francesca Toni Paolo Torroni

January 9, 2003

DEIS Technical Report no. DEIS-LIA-02-008 *LIA Series no. 61*

A multi-stage negotiation architecture for sharing resources amongst logic-based agents

Fariba Sadri [1] Francesca Toni [1] Paolo Torroni [2]

[1] *Department of Computing*
Imperial College
180 Queen's Gate
SW7 2BZ London, UK
{fs,ft}@doc.ic.ac.uk

[2] *DEIS*
Università di Bologna
V.le Risorgimento 2
40136 Bologna, Italy
ptorroni@deis.unibo.it

January 9, 2003

Abstract. In earlier work we proposed a framework for agents negotiating for resources. The framework was based on abductive logic programming. Agents had knowledge bases, goals and plans for achieving their goals. The plans required resources, and each agent did not necessarily own at the start all the resources it required. Agents had to negotiate with each other to obtain the resources that they needed. In this paper we extend the earlier work by providing the agents with richer knowledge and more sophisticated negotiation capabilities so that they can negotiate periods of time during which they would have use of resources, thus allowing for the sharing of resources. We explore a sequence of stages of negotiation such that in the earlier stages agents need to do little "thinking" and disclose very little about their plans and constraints, but as they step through the sequence they will have to do more "thinking" and disclose more about themselves. If one stage fails to produce a deal amongst the agents they may agree to move to the next stage where there is a better chance of a mutually agreeable deal.

Keywords: *Multi-agent systems, negotiation, logic based reasoning, abduction, resource exchange*

1 Introduction

In earlier work [5, 7, 6] we proposed a framework for agents negotiating for resources. The framework was based on abductive logic programming. The knowledge of the agents consisted of a tuple $< \mathcal{B}, \mathcal{R}, \mathcal{I}, \mathcal{D}, \mathcal{G} >$ with

- $\mathcal{B}$: *beliefs* about the world, such as the information about the self and the partner agents, and the negotiation policies of the agent;

- $\mathcal{R}$: initially owned *resources*;

- $\mathcal{I}$: *intentions*, i.e., the plan that the agent intends to carry out, in order to achieve his goals and the required resources for that plan;

- $\mathcal{D}$: store of past *dialogues*; and

- $\mathcal{G}$: agent *goals*.

It was assumed that all the missing resources in the plan in $\mathcal{I}$ would have to be acquired at the start before the plan could be carried out, but there was no explicit reference to time windows for actions and durations of actions in the plan. Furthermore, it was assumed that once a resource was acquired by an agent it was his indefinitely.

In this paper we relax these assumptions to provide a richer framework for negotiation that allows better sharing of resources. We allow agent plans to be partially ordered sets of activities specified within time windows and with each activity of the plan specifying its required resources. An activity a in a plan can be specified as $< a, R_a, D_a, Ts_a, Te_a >$ denoting that activity a has duration D_a, earliest start time Ts_a, latest end time Te_a, and requires resource R_a. Without loss of generality, we assume that each activity needs only one resource. It is possible to model activities requiring multiple resources by putting together more activities to be carried out in parallel.

These extensions enable us to develop negotiation policies that allow agents to exchange resources and strike deals for agreed time windows so that (non-consumable) resources can be shared, thus allowing solutions for a wider range of resource reallocation problems.

We introduce a multi-stage process of negotiation, where a higher stage involves more information passing between agents and more sophisticated negotiation and possible re-planning by the agents aimed at trying to help satisfy each other's constraints and requirements.

To introduce the ideas consider the following example.

Example 1 There are two agents, x and y. a is an activity in x's intentions, i.e. $< a, r, 3, 1, 5 > \in \mathcal{I}_a$. y has in its intentions an activity b which also requires r. x has r from time 1 to 3 and from time 5 onwards, and y has r from time 3 to 5. Agent x needs to obtain r for at least 3 consecutive time slots between 1 and 5, but he currently has r only for 2 time slots. Let us consider three different scenarios for y's intentions.

1. y's activity is to be carried out between 4 and 10, and lasts 1 time unit: $< b, r, 1, 4, 10 > \in \mathcal{I}_b$. The problem will be solved by y agreeing to give r to x for $[3, 4]$.

2. $< b, r, 1, 3, 10 > \in \mathcal{I}_b$ and suppose y has an initial concrete schedule for b to be carried out between 3 and 4. The problem is solved by y agreeing to postpone his schedule by 1 time slot and giving r to x for $[3, 4]$.

3. $< b, r, 2, 3, 10 > \in \mathcal{I}_b$. 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 $[5, 6]$. $\square$

In this paper, we first define the problem of resource reallocation with time constraints, then we introduce the multi-stage negotiation and define three different stages and their properties. We end with a discussion.

2 Background and Preliminaries

2.1 Background

In this section we review some concepts from our earlier papers needed for this paper.

Definition 1 An *agent system* is a finite set $\mathcal{S}$, with at least two elements, where each $A \in \mathcal{S}$ is a ground term, representing the name of an agent, and each agent is equipped at each time T with a *knowledge base* $\mathcal{K}_{A,T}$. All elements of $\mathcal{S}$ are distinct (agent names must be uniquely identifiable).

Definition 2 Given an agent system $\mathcal{S}$, an agent $x \in \mathcal{S}$, and a natural number $\tau \in \mathbb{N}$, x's *knowledge base* $\mathcal{K}_{x,\tau}$ is defined as a tuple $< \mathcal{B}_x, \mathcal{R}_x, \mathcal{I}_x, \mathcal{D}_{x,\tau}, \mathcal{G}_x, \hat{\mathcal{I}}_x >$, where the meaning of the symbols $\mathcal{B}$, $\mathcal{R}$, $\mathcal{I}$, $\mathcal{D}$ and $\mathcal{G}$ in the tuple is the one defined in the introduction, and $\hat{\mathcal{I}}_x$, which is a new component, is a (possibly empty) concrete schedule for the activities in $\mathcal{I}_x$. Thus, if $< a, t_s, t_e > \in \hat{\mathcal{I}}_x$ then $< a, r, d, t_{s_a}, t_{e_a} > \in \mathcal{I}_x$ and $t_e - t_s + 1 = d$ and $t_s \geq t_{s_a}$ and $t_e \geq t_{e_a}$.

In this paper, we assume that the only part of $\mathcal{K}$ that changes is the dialogue store $\mathcal{D}$ which grows in time. We assume also that in $\mathcal{I}_x$ all activities that require the same resource have disjoint time windows.

We assume that all agents in a system share the same language, in terms of syntax, semantics, ontology, and pragmatics. We refer to [6] 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 utterance (see below for a concrete language).

Definition 3 *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 dialogue moves.

The intended use of these policies is that if the agent receives a move p_i, and the conditions C are "satisfied" (represented by an entailment operator $\vdash$) in its knowledge base, it generates p_{i+1}. In [6] we have offered the operational model of policies. In this paper we assume all agents in a system share the same policies.

Definition 4 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 $\mathcal{L}$, $\{p_0, p_1, p_2, \dots\}$, such that, for a given set of time lapses $0 \leq \tau_0 < \tau_1 < \tau_2 < \dots$:

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, $p_{i-1} \wedge C \Rightarrow [p_i \wedge t_{i-1} < t_i]$ s.t. $\mathcal{K}_{U,t_i} \wedge p_{i-1} \vdash C$;
4. there is an identifier D such that, $\forall\, i \geq 0$, the dialogue identifier of p_i is D;

A dialogue $\{p_0, p_1, \dots p_m\}$, $m \geq 0$, is *terminated* if p_m is a ground final move, namely p_m is a ground instance of an utterance in $\mathcal{F}(\mathcal{L})$.

The rest of this section introduces new concepts, to be used in the main body of the paper.

Definition 5 Let $\mathcal{S}$ be an agent system. Let $\mathcal{R}es$ be a set of resources.[1] 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 in a system $\mathcal{S}$ of agents as follows: $T_{max} \stackrel{def}{=} \max_{x \in \mathcal{S}}\{TE \mid \; < \_, \_, \_, \_, TE > \in \mathcal{I}_x\}$.

The negotiation process modifies the resource distribution in the agent system through time. The *Temporal Resource Reallocation Problem (T-RRP)* is the problem of finding an answer 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

[1] 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).

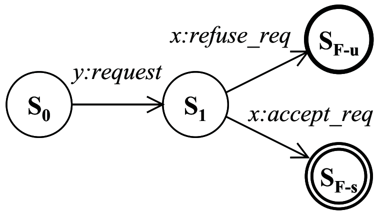

Figure 1. *Protocol for Stage1 and Stage2*

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.

The communication language, called $\mathcal{L}_{1-TW}$, adopted in this paper, extends $\mathcal{L}_1$ in [7] by adding further dialogue moves and time windows.

Definition 6 $\mathcal{L}_{1-TW} = \{tell(X, Y, Content, D, T) | Content \in$
$\{$ (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 $Ts' < Te'$ for (5) and (6).

The first three moves are of intuitive meaning. (4) is used by X to propose a deal: 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, (6) is used to accept a deal. 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).

A dialogue protocol is 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 utterances. When we show the protocols, we use some abbreviations:

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

and so forth. An example of protocol is given in Fig. 1.

An important property of policies used to induce dialogues as in Definition 2.1 is *conformance* to protocols. In order to define the concept of conformance, we define dialogues in relation with protocols.

Definition 7 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 $\mathcal{L}$, $\{p_0, p_1, p_2, \dots\}$, such that, for a given set of time lapses $0 \le \tau_0 < \tau_1 < \tau_2 < \dots$:

1. $\forall\ i \ge 0$, p_i is uttered at time τ_i;
2. $\forall\ i \ge 0$, if p_i is uttered by agent X (viz. Y), then p_{i+1} (if any) is uttered by agent Y (viz. X);
3. the content of p_0 must label an arc from S_0. $\forall i$, if the content of p_i is a 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;
4. there is an identifier D such that, $\forall\ i \ge 0$, the dialogue identifier of p_i is D.

A dialogue $\{p_0, p_1, \dots p_m\}$, $m \ge 0$, 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 8 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 9 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, $\sigma = \{d_0, d_1, d_2, \dots\}$, where $d_j = \{p_{j0}, p_{j1}, \dots, p_{jn}\}$ for all j, such that, for a given set of time lapses $0 \le \tau_0 < \tau_1 < \tau_2 < \dots$:

1. $\forall\, i \ge 0$, d_i is initiated by X at time τ_i, and it is terminated at a time $\tau_i' < \tau_{i+1}$;
2. $\nexists i, j, i \ne j$ such that $p_{i0} = p_{j0}$;
3. $\forall i$, if d_i and d_{i+1} are in σ, then $d_i = \{p_{i0}, \dots, p_{in}\}$ and $p_{in} = \mathit{tell}(Y, X, \mathit{refuse}(\mathit{request}(\ \mathit{give}(R,\ (Ts, Te)\ ))), d_i,\ \tau)$ for some Ts, Te, τ;

4. $\exists < A, R, D, TS, TE > \in \mathcal{I}_X$ such that $\forall i$ if $d_i \in \sigma$ then $p_{i0} = tell(X, Y, request($ $give(R, (Ts, Te)\ )), d_i, \tau_i))$ for some Ts, Te such that $Ts \geq TS$ and $Te \leq TE$;

5. $\forall i, d_i = \{p_{i0}, \ldots, p_{in}\}$ is the last dialogue in the sequence if for some $\tau, T_s, T_e, T'_s, T'_e$, $p_{in} = tell(Y, X, accept(\ request(\ give(R, (Ts, Te)\ ))), \tau)$ or $p_{in} = tell(Y, X, accept($ $promise(R, (Ts, Te), (Ts', Te')\ )), d_i, \tau)$.

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 if they are participating in an ongoing sequence of dialogues which they initiated (an agent cycle very similar to the one in [6] will achieve this atomicity).

3 Negotiation Stages

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

1. Request/rigid schedule
2. Request/flexible schedule
3. Blind deal

In this section we briefly define them and discuss their differences. For each stage we define (*i*) the *protocol*, (*ii*) the *policies* adopted by both agents, and (*iii*) 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, and (*iv*) subsumption of earlier stages (in terms of solvable problems).

For each stage we give an example of a problem that can be solved within it, and an example that cannot.

3.1 Stage 1 - request/rigid schedule

Stage 1 negotiation is a conservative extension of the two-step negotiation scheme used in [6]. In particular, Stage 1 extends [6] by introducing time windows, and thus allowing for resource sharing, but the main features of the policy do not change (it reflects the behaviour of a self-interested agent).

$$
\begin{array}{ll}
(IC.1) & \mathit{tell}(X, y, \mathbf{request}(\mathbf{give}(R, (Ts, Te))), D, T) \\
& \wedge\ \mathit{have}(R, (Ts, Te), T) \wedge \neg\ \mathit{need1}(R, (Ts, Te), T) \\
& \qquad \Rightarrow\ \exists\, T' \mid \mathit{tell}(y, X, \mathbf{accept}(\mathbf{request}(\mathbf{give}(R, (Ts, Te)))), D, T') \wedge T < T' \\
(IC.2) & \mathit{tell}(X, y, \mathbf{request}(\mathbf{give}(R, (Ts, Te))), D, T) \\
& \wedge\ \mathit{have}(R, (Ts, Te), T) \wedge \mathit{need1}(R, (Ts, Te), T) \\
& \qquad \Rightarrow \exists\, T' \mid \mathit{tell}(y, X, \mathbf{refuse}(\mathbf{request}(\mathbf{give}(R, (Ts, Te)))), D, T') \wedge T < T' \\
(IC.3) & \mathit{tell}(X, y, \mathbf{request}(\mathbf{give}(R, (Ts, Te))), D, T) \\
& \wedge\ \neg\ \mathit{have}(R, (Ts, Te), T) \\
& \qquad \Rightarrow \exists\, T' \mid \mathit{tell}(y, X, \mathbf{refuse}(\mathbf{request}(\mathbf{give}(R, (Ts, Te)))), D, T') \wedge T < T'
\end{array}
$$

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

3.1.1 Protocol

The protocol is shown in Fig. 1. The allowed dialogue moves are those with content (1), (2), (3) in $\mathcal{L}_{1-TW}$. A *request*(*give*$(R, (Ts, Te))$) is made in order to obtain a resource R from time Ts to time Te. The agent receiving the request can either answer positively (*accept*) or negatively (*refuse*). The first answer leads to a successful final state, the second one leads to an unsuccessful final state.

3.1.2 Policy

The policy is shown in Fig. 2. It is based on the assumption that the answering agent (y) has a concrete schedule of activities, $\hat{\mathcal{I}}_y$. y answers positively (*accept*) if it can give R away for the requested time window, without modifying its schedule, and answers no otherwise (*refuse*). In the Appendix we give the definitions of all auxiliary predicates used in the policies and the results below. Intuitively, $\mathit{have}(R, (Ts, Te), T)$ indicates that at time T the agent knows that it has resource R for the period of time (Ts, Te), possibly as a result of past negotiation dialogues, and $\mathit{need1}(R, (Ts, Te), T)$ means that at time T the agent knows that it has the resource for the period (Ts, Te), and has a scheduled activity that makes use of R during part of that period. We assume that the agent will prove in finite time whether an atom is satisfied in its knowledge base, for all predicates defined in the Appendix. The symbol $\neg$ has the semantics of negation by failure ($\neg\ p$ is true if the agent fails to prove p).

Example 2 Let us consider a system of two agents, x and y, two activities a and b, and a resource r. Let us assume that x has activity b and y has activity a, and the initial resource assignment is: r to y, from 10 to 20. Both activities need r for five hours and can be carried out between 10 and 20, and y has decided to do activity a from 10 to 15:

$$
\begin{array}{ll}
\mathcal{I}_y = \{< a, r, 5, 10, 20 >\} & \mathcal{I}_x = \{< b, r, 5, 10, 20 >\} \\
\hat{\mathcal{I}}_y = \{< a, r, 5, 10, 15 >\} & \hat{\mathcal{I}}_x = \emptyset \\
\mathcal{R}_y = \{(r, (10, 20))\} & \mathcal{R}_x = \emptyset
\end{array}
$$

This problem cannot be solved without resource sharing. In particular, it cannot be solved by the approach of [7], which requires that a resource is given away forever. Instead, it can be solved by the following dialogue $d(1)$ occurring at Stage 1:

$tell(x, y, request(give(r, (15, 20))), d(1), 1)$

$tell(y, x, accept(request(give(r, (15, 20)))), d(1), 2)$

If the operational model adopted by x allows for multiple requests of r with different time windows, we can imagine that $d(1)$ is the last of a (successful) sequence of dialogues started by x requesting r. □

3.1.3 Properties

(i): compliance of the policy to the protocol. It is easy to prove that the policy represented in Fig. 2 is compliant with the protocol in Fig. 1. In addition, the program constituted by the policy and the relevant definitions for *have* and *need1* is exhaustive and deterministic [7], therefore it is possible to use it to directly implement a negotiation framework, based on an operational model such as that proposed in [6].

(ii): properties of single dialogues. A request/rigid schedule interaction enjoys the property of termination. Dialogues following this protocol have a fixed number of steps (2).

The problems that can be solved at this stage are those that can be solved without the need for the agent receiving a request to modify its current activity scheduling. Formally:

Theorem 1 Let us consider a system composed of two agents, x and y, with y (at least) having a concrete schedule and both having an initial resource assignment. Then, for all resources r in the system, all times τ, all activities a assigned to x and all intervals $[Ts, Te]$ such that

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

$\mathcal{K}_{y,\tau} \vdash available1(r, (Ts, Te))$,

there exists a dialogue d induced by the policy of Stage 1, starting at time τ, and ending at some time τ', such that

$\mathcal{K}_{x,\tau'} \vdash have(r, (Ts, Te), \tau') \quad \wedge$

$\mathcal{K}_{y,\tau'} \vdash indifferent(r, (Ts, Te))$

Based on these results, we argue that Stage 1 is a conservative extension of the work done in [6]. In particular, the $\mathcal{N}$-agents of [6] have been proved *weak complete* [7], that is, able to produce a dialogue that solves a resource reallocation problem, whenever there are sufficient resources in the system for the activities that require them. Stage 1 is able to solve all problems solvable by $\mathcal{N}$-agents, and it is therefore also *weak complete*, but it is also able to solve some problems that $\mathcal{N}$-agents cannot solve, such as Example 2.

(iii): **properties of sequences of dialogues.** It is up to the agent who is missing a resource to find good heuristics to formulate a request, or a series of 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 $\mathcal{K}_{x,\tau} \vdash miss(r, (Ts, Te), a)$, until all such intervals are exhausted, or the request for one of the intervals is accepted. We call this strategy *Strategy1*. Th. 2 shows under what circumstances Stage 1 is complete:

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 his attempt to acquire r from y, starting at τ with a request for r for an interval $[Ts, Te]$ such that $\mathcal{K}_{x,\tau} \vdash miss(r, (Ts, Te), a)$, then the resulting sequence of dialogues will be finite and will terminate at a time τ', and either

$\mathcal{K}_{x,\tau'} \vdash feasible2(a, (Ts^\star, Te^\star))$, for some $[Ts^\star, Te^\star]$, or $\nexists\, Ts', Te'$ such that
$\mathcal{K}_{x,\tau} \vdash miss(r, (Ts', Te'), a) \wedge$
$\mathcal{K}_{y,\tau} \vdash available1(r, (Ts', Te'))$.

This result follows from the fact that time is discrete and T_{max} is finite, therefore, for a given r and a, the set of time periods such that $\mathcal{K}_{x,\tau} \vdash miss(r, (Ts, Te), a)$ is finite.

(iv): **subsumption of earlier stages.** Stage 1 subsumes the work done in [6], in the sense that the resource reallocation problems it solves include and extend those solved by [6].

We now give an example that cannot be solved at this stage but that has a solution at the next stage.

Example 3 Let us consider the following modification of Example 2, where y has a different initial schedule:

$$\begin{array}{ll} \mathcal{I}_y = \{< a, r, 5, 10, 20 >\} & \mathcal{I}_x = \{< b, r, 5, 10, 20 >\} \\ \hat{\mathcal{I}}_y = \{< a, r, 5, 11, 16 >\} & \hat{\mathcal{I}}_x = \emptyset \\ \mathcal{R}_y = \{(r, (10, 20))\} & \mathcal{R}_x = \emptyset \end{array}$$

There is no solution if y sticks to $\hat{\mathcal{I}}_y$: but if y was happy with a different schedule, e.g. by scheduling a in $[15, 20]$ rather than in $[11, 16]$, then x and y could both do their activities by sharing resource r. $\square$

3.2 Stage 2 - request/flexible schedule

3.2.1 Protocol

The protocol is the same as for Stage 1.

$$
\begin{array}{ll}
(IC.1') & tell(X, y, \mathbf{request}(\mathbf{give}(R, (Ts, Te))), D, T) \\
& \wedge have(R, (Ts, Te), T) \wedge \neg\ need2(R, (Ts, Te), T) \\
& \qquad \Rightarrow\ \exists\ T' \mid tell(y, X, \mathbf{accept}(\mathbf{request}(\mathbf{give}(R, (Ts, Te)))), D, T') \wedge T < T' \\
(IC.2') & tell(X, y, \mathbf{request}(\mathbf{give}(R, (Ts, Te))), D, T) \\
& \wedge have(R, (Ts, Te), T) \wedge need2(R, (Ts, Te), T) \\
& \qquad \Rightarrow \exists\ T' \mid tell(y, X, \mathbf{refuse}(\mathbf{request}(\mathbf{give}(R, (Ts, Te)))), D, T') \wedge T < T' \\
(IC.3) & tell(X, y, \mathbf{request}(\mathbf{give}(R, (Ts, Te))), D, T) \\
& \wedge \neg\ have(R, (Ts, Te), T) \\
& \qquad \Rightarrow \exists\ T' \mid tell(y, X, \mathbf{refuse}(\mathbf{request}(\mathbf{give}(R, (Ts, Te)))), D, T') \wedge T < T'
\end{array}
$$

Figure 3. Stage 2 *policy for an agent* y

3.2.2 Policy

The policy is shown in Fig. 3. It is *not* based on the assumption that the agent to whom a *request* is being addressed has any concrete schedule of activities. In fact, as opposed to the previous stage, agents negotiating in Stage 2 adopt a different semantics for "need" (*need2*) which does not refer to their $\hat{\mathcal{I}}$.

As a consequence, an agent will *accept* a *request* for R if it can find a scheduling 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.

Example 3 can be solved by the following negotiation dialogue $d(2)$ occurring at Stage 2:

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

3.2.3 Properties

In comparison to Stage 1, Stage 2 is computationally more demanding for the agent who is replying to a request. The problems that can be solved at this stage are all those solved by Stage 1, plus all those that can be solved by means of a modification in the agents' current activity scheduling. Again, it is up to the requesting agent to find good heuristics to formulate a request or a series of requests which can be accepted.

(i): compliance of the policy to the protocol. It is possible to prove that the policy represented in Fig. 3 is compliant to the protocol in Fig. 1. In addition, the policies together with the auxiliary predicate definitions are exhaustive and deterministic.

(ii): **properties of single dialogues.** Since the protocol is the same as for Stage 1, a request/flexible schedule interaction enjoys the property of termination. In fact, dialogues following this protocol have a fixed number of steps (2).

Theorem 3 Let us consider a system composed of two agents, x and y, each having an intention $\mathcal{I}$ and an initial resource assignment. Then, for all resources r in the system, all activities a assigned to x, all times τ, and all intervals $[Ts, Te]$ s.t.

$$\mathcal{K}_{x,\tau} \vdash \textit{miss}(r, (Ts, Te), a) \quad \wedge$$
$$\mathcal{K}_{y,\tau} \vdash \textit{available2}(r, (Ts, Te)),$$

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

$$\mathcal{K}_{x,\tau'} \vdash \textit{need2}(r, (Ts, Te), a, \tau') \quad \wedge$$
$$\mathcal{K}_{y,\tau'} \vdash \textit{indifferent}(r, (Ts, Te))$$

(iii): **properties of sequences of dialogues.** It is possible to prove the following result:

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 that require r, if x follows *Strategy1* in his 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 the resulting sequence of dialogues will be finite and will terminate at a time τ', and either

$\mathcal{K}_{x,\tau'} \vdash \textit{feasible2}(a, (Ts^\star, Te^\star))$, for some $[Ts^\star, Te^\star]$, or $\nexists\, Ts', Te'$ such that
$\mathcal{K}_{x,\tau} \vdash \textit{miss}(r, (Ts', Te'), a) \wedge$
$\mathcal{K}_{y,\tau} \vdash \textit{available2}(r, (Ts', Te'))$.

(iv): **subsumption of earlier stages.** Stage 2 subsumes Stage 1. This follows from the definition of *feasible1* and *feasible2*:

Theorem 5 Let x be an agent in a system, and $\mathcal{K}_{x,\tau}$ its knowledge at time τ. Then, for all activities a and times Ts, Te,

$\mathcal{K}_{x,\tau} \vdash \textit{feasible1}(a, (Ts, Te)) \Rightarrow \mathcal{K}_{x,\tau} \vdash \textit{feasible2}(a, (Ts, Te))$

Corollary 6 Let us consider a system composed of two agents, x and y, each having an intention $\mathcal{I}$ and a resource assignment. Then, for all resources r in the system, all activities a assigned to x, all concrete schedules $\hat{\mathcal{I}}_y$, all times τ and all intervals $[Ts, Te]$ such that

$$\mathcal{K}_{x,\tau} \vdash \textit{miss}(r, (Ts, Te), a) \quad \wedge$$
$$\mathcal{K}_{y,\tau} \vdash \textit{available1}(r, (Ts, Te)),$$

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

$$\mathcal{K}_{x,\tau'} \vdash \textit{have}(r, (Ts, Te), \tau'),$$
$$\mathcal{K}_{y,\tau'} \vdash \textit{indifferent}(r, (Ts, Te)).$$

We will now give a counterexample for Stage 2.

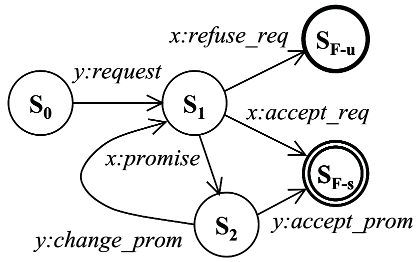

Figure 4. *Protocol for Stage3*

Example 4 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 5):

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

We did not give any concrete initial schedules for the agents' activities since it plays a role only at Stage 1. There is no solution to this problem that can be found at Stage 2. In fact, if y gives away r, it will not be possible for it to carry out a any more. □

3.3 Stage 3 - blind deal

3.3.1 Protocol

The protocol is shown in Fig. 4. The allowed dialogue moves are those of the previous stages plus those with content:

$promise(R, (Ts', Te'), (Ts, Te))$;
$change(promise(R, (Ts', Te'), (Ts, Te)))$;
$accept(promise(R, (Ts', Te'), (Ts, Te)))$.

After the initial move, $request(give(R, (Ts, Te)))$, which makes the dialogue reach state S_1, the other agent can either *accept* or *refuse*, as in Stages 1 and 2, or can propose a deal (*promise*). After a promise, the agent who made the request can either accept the deal, causing a successful termination, or ask for a new promise, which brings the state back to S_1.

3.3.2 Policy

The policy is shown in Fig. 5.[2] An agent will *accept* a request about a resource R for the period $[Ts, Te]$ if it would do it at Stage 2, but it will refuse it *only if* it does not

[2] 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, and – with abuse of notation – $\mathcal{K}_{X,\tau} \cup \Delta \vdash p$, where Δ is a set of atoms, to mean that

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 there exist such intervals), until y accepts one (successful termination), or there exist no new ones to propose, in which case x will refuse the initial request, which leads 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 3 a predicate $pick((Ts, Te), T)$, that is a deterministic and exhaustive predicate, that at any given time T uniquely determines a time period having made reference possibly to $\mathcal{D}_{x,T}$ and $\mathcal{R}_x$. By exhaustive we mean that if a *promise* is possible, then one will always be found.

Example 4 can be solved by the following negotiation dialogue $d(3)$ occurring at Stage 3:

$tell(x, y, request(give(r, (10, 15))), d(3), 1)$
$tell(y, x, promise(r, (15, 20), (10, 15)), d(3), 2)$
$tell(x, y, accept(promise(r, (15, 20), (10, 15))), d(3), 3)$

3.3.3 Properties

At Stage 3 we require that agents are more cooperative than at Stages 2 and 1. This is achieved by both agents – and in particular the agent to whom the request is addressed – to do more reasoning in order to be helpful in response to requests.

(i): **compliance of the policy to the protocol.** It is possible to prove that the policy represented in Fig. 5 is compliant to the protocol in Fig. 4. In addition, the policies together with the definitions are exhaustive and deterministic.

(ii): **properties of single dialogues.** A *blind deal* interaction does not enjoy the fixed dialogue length property as in the previous two stages, but it terminates if we have a finite scheduling horizon T_{max} and we do not allow the same utterance to be made twice at different times, during the same dialogue (which is a reasonable requirement). It is computationally more demanding than *request/flexible schedule* for both agents.

$\mathcal{K}_{X,\tau}$, enlarged with Δ, entails p. For instance, $\mathcal{K}_{X,\tau} \vdash have(R, (Ts, Te), \tau)$ means that, according to X's knowledge base as at time τ, X has (or had or will have) R (continuously) from Ts to Te.

$$
\begin{aligned}
(IC.1')\quad & \mathit{tell}(X, y, \mathbf{request}(\mathbf{give}(R, (Ts, Te))), D, T) \\
& \wedge \mathit{have}(R, (Ts, Te), T) \ \wedge \neg \ \mathit{need2}(R, (Ts, Te), T) \\
& \qquad \Rightarrow \ \exists\, T' \mid \mathit{tell}(y, X, \mathbf{accept}(\mathbf{request}(\mathbf{give}(R, (Ts, Te)))), D, T') \wedge T < T'
\end{aligned}
$$

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

$$
\begin{aligned}
(IC.4)\quad & \mathit{tell}(X, y, \mathbf{request}(\mathbf{give}(R, Ts, Te)), D, T) \\
& \wedge \ \mathit{have}(R, (Ts, Te), T) \\
& \wedge \ \mathit{need2}(R, (Ts, Te), A, T) \\
& \wedge \nexists\, Ts', Te' \mid \neg \ \mathit{have}(R, (Ts', Te')) \\
& \wedge \ \mathit{tell}(y, X, \mathit{promise}(R, (Ts', Te'), (Ts, Te)), D, \_) \notin \mathcal{D}_{X,T} \\
& \wedge \mathcal{K}_{X,T} \cup \{\mathit{obtain}(R, (Ts', Te')), \mathit{give\_away}(R, (Ts, Te))\} \ \vdash \ \mathit{feasible2}(A, (Ts', Te')) \\
& \qquad \Rightarrow \exists\, T' \mid \mathit{tell}(y, X, \mathbf{refuse}(\mathbf{request}(\mathbf{give}(R, Ts, Te))), D, T') \wedge T < T'
\end{aligned}
$$

$$
\begin{aligned}
(IC.5)\quad & \mathit{tell}(X, y, \mathbf{request}(\mathbf{give}(R, Ts, Te)), D, T) \\
& \wedge \ \mathit{have}(R, (Ts, Te), T) \ \wedge \ \mathit{need2}(R, (Ts, Te), A, T) \\
& \wedge \ \mathit{pick}((Ts', Te'), T) \wedge \ \neg \ \mathit{have}(R, (Ts', Te'), T) \\
& \wedge \ \mathit{tell}(y, X, \mathit{promise}(R, (Ts', Te'), (Ts, Te)), D, \_) \notin \mathcal{D}_{X,T} \\
& \wedge \mathcal{K}_{X,T} \cup \{\mathit{obtain}(R, (Ts', Te')), \mathit{give\_away}(R, (Ts, Te))\} \ \vdash \ \mathit{feasible2}(A, (Ts', Te')) \\
& \qquad \Rightarrow \exists\, T' \mid \mathit{tell}(y, X, \mathbf{promise}(R, (Ts', Te'), (Ts, Te)), D, T') \wedge T < T'
\end{aligned}
$$

$$
\begin{aligned}
(IC.6)\quad & \mathit{tell}(X, y, \mathbf{change}(\mathbf{promise}((\_, \_), (Ts, Te))), D, T) \\
& \wedge \ \mathit{have}(R, (Ts, Te), T) \ \wedge \ \mathit{need2}(R, (Ts, Te), A, T) \\
& \wedge \ \mathit{pick}((Ts', Te'), T) \wedge \ \neg \ \mathit{have}(R, (Ts', Te'), T) \\
& \wedge \ \mathit{tell}(y, X, \mathit{promise}(R, (Ts', Te'), (Ts, Te)), D, \_) \notin \mathcal{D}_{X,T} \\
& \wedge \mathcal{K}_{X,T} \cup \{\mathit{obtain}(R, (Ts', Te')), \mathit{give\_away}(R, (Ts, Te))\} \ \vdash \ \mathit{feasible2}(A, (Ts', Te')) \\
& \qquad \Rightarrow \exists\, T' \mid \mathit{tell}(y, X, \mathbf{promise}(R, (Ts', Te'), (Ts, Te)), D, T') \wedge T < T'
\end{aligned}
$$

$$
\begin{aligned}
(IC.7)\quad & \mathit{tell}(X, y, \mathbf{change}(\mathbf{promise}(R, (Ts', Te'), (Ts, Te))), D, T) \\
& \wedge \ \mathit{have}(R, (Ts, Te), T) \ \wedge \ \mathit{need2}(R, (Ts, Te), A, T) \\
& \wedge \nexists\, Ts', Te' \mid \neg \ \mathit{have}(R, (Ts', Te'), T) \\
& \wedge \ \mathit{tell}(y, X, \mathit{promise}(R, (Ts', Te'), (Ts, Te)), D, \_) \notin \mathcal{D}_{X,T} \\
& \wedge \mathcal{K}_{X,T} \cup \{\mathit{obtain}(R, (Ts', Te')), \mathit{give\_away}(R, (Ts, Te))\} \ \vdash \ \mathit{feasible2}(A, (Ts', Te')) \\
& \qquad \Rightarrow \exists\, T' \mid \mathit{tell}(y, X, \mathbf{refuse}(\mathbf{request}(\mathbf{give}(R, (Ts, Te)))), D, T') \wedge T < T'
\end{aligned}
$$

$$
\begin{aligned}
(IC.8)\quad & \mathit{tell}(X, y, \mathbf{promise}(R, (Ts', Te'), (Ts, Te)), D, T) \ \wedge \ \mathit{miss}(R, (Ts, Te), A) \\
& \wedge \mathcal{K}_{X,T} \ \cup \ \{\mathit{obtain}(R, (Ts, Te)), \mathit{give\_away}(R, (Ts', Te'))\} \ \vdash \ \mathit{feasible2}(A, (\_, \_)) \\
& \qquad \Rightarrow \exists\, T' \mid \mathit{tell}(y, X, \mathbf{accept}(\mathbf{promise}(R, (Ts', Te'), (Ts, Te)), D, T')) \\
& \qquad \wedge T < T'
\end{aligned}
$$

$$
\begin{aligned}
(IC.9)\quad & \mathit{tell}(X, y, \mathbf{promise}(R, (Ts', Te'), (Ts, Te)), D, T) \ \wedge \ \mathit{miss}(R, (Ts, Te), A) \\
& \wedge \mathcal{K}_{X,T} \ \cup \ \{\mathit{obtain}(R, (Ts, Te)), \mathit{give\_away}(R, (Ts', Te'))\} \ \nvdash \ \mathit{feasible2}(A, (\_, \_)) \\
& \qquad \Rightarrow \exists\, T' \mid \mathit{tell}(y, X, \mathbf{change}(\mathbf{promise}(R, (Ts', Te'), (Ts, Te)), D, T')) \\
& \qquad \wedge T < T'
\end{aligned}
$$

Figure 5. Stage 3 *policy for an agent* y

Theorem 7 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

$$\mathcal{K}_{x,\tau} \vdash \mathit{miss}(r, (Ts, Te), a) \quad \wedge$$
$$\mathcal{K}_{y,\tau} \vdash \mathit{need2}(r, (Ts, Te), b) \quad \wedge$$
$$\exists [Ts', Te'] \text{ such that } \mathcal{K}_{y,\tau} \cup \{\mathit{give\_away}(r, (Ts, Te)),$$
$$\mathit{obtain}(r, (Ts', Te'))\} \vdash \mathit{feasible2}(b, (Ts'', Te''))$$

for some $[Ts'', Te'']$, there exists a dialogue d induced by the policy of Stage 3, starting at time τ, and ending at a time τ', such that

$$\mathcal{K}_{x,\tau'} \vdash \mathit{need2}(r, (Ts, Te), a) \quad \wedge$$
$$\mathcal{K}_{y,\tau'} \vdash \mathit{indifferent}(r, (Ts, Te))$$

(iii)**: properties of sequences of dialogues.** Similarly to the previous stages, it is possible to prove the following result:

Theorem 8 Let us consider a system composed of two agents, x and y, negotiating at Stage 3. Then, for all resources r in the system, and all activities a assigned to x which require r, if starting at time τ, x successively requests r for the time intervals $[Ts, Te]$ such that $\mathcal{K}_{x,\tau} \vdash \mathit{miss}(r, (Ts, Te), a)$, the sequence of dialogues will terminate at τ' after a finite number of dialogues, and either

$$\mathcal{K}_{x,\tau'} \vdash \mathit{feasible2}(a, (Ts^\star, Te^\star))$$

for some $[Ts^\star, Te^\star]$, or $\nexists\ [Ts, Te], [Ts', Te']$ such that

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

(iv)**: subsumption of earlier stages.** More problems can be solved by Stage 3 than by each of the previous stages. The following formalizes this result.

Corollary 9 Let us consider a system composed of two agents, x and y, each having an intention $\mathcal{I}$ and an initial resource assignment. Then, for all resources r in the system, all activities a assigned to x, all times τ, and all intervals $[Ts, Te]$ such that

$$\mathcal{K}_{x,\tau} \vdash \mathit{miss}(r, (Ts, Te), a) \quad \wedge$$
$$\mathcal{K}_{y,\tau} \vdash \mathit{available2}(r, (Ts, Te)),$$

there exists a dialogue d **induced by the policy of Stage 3**, starting at time τ, and ending at time τ', such that

$$\mathcal{K}_{x,\tau'} \vdash \mathit{need2}(r, (Ts, Te), a) \quad \wedge$$
$$\mathcal{K}_{y,\tau'} \vdash \mathit{indifferent}(r, (Ts, Te))$$

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. Let us consider the following counterexample for Stage3.

Example 5 Let us consider the following modification of Example 2, where there is a different resource and activity assignment:

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

There is no solution to this problem that can be found at Stage 3. In fact, the (minimal) requests that y 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

In this paper we presented an approach to resource reallocation with temporal constraints, for self-interested, co-operative agents. We introduced a multi-stage process of negotiation, where a higher stage would involve more computation and more sophisticated negotiation and possible re-planning by the agents aimed at trying to help satisfy each other's constraints and requirements.

This paper extends our earlier work on multi-agent negotiation for resources [5, 7, 6] by allowing more sophisticated negotiation to satisfy more flexible time constrained partially ordered activities. It shares with the earlier work the logic-based high level framework that facilitates specification and proof of formal properties and which has an operational model, that forms a bridge between the system description and implementation.

Our modelling of constraints draws inspiration from work done by H.H. El Sakkout and M. Wallace on constraint satisfaction for a monolithic system [2], although it is more simplified. But our work differs from [2] 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 predefined schedules. An extension of [2] to multi-agent systems has been proposed by C.A. Harrison in [3], but the focus there is again that of minimizing changes to existing schedules, and not on negotiation.

[1] proposes an approach to negotiation based on multiple stages, focusing on degree of coordination. 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 stage it is possible, for instance, to discover that the overall problem is over-constrained and therefore a certain goal is infeasible.

More recent work in problem solving in presence of temporal constraints in a multi-agent setting is that of [8], where the author models other agents' constraints by means of a "shadow" constraint store to solve a meeting scheduling problem. The approach adopts a basic protocol (*propose-accept/reject*) and a cycle which is very similar to that of [6], and uses heuristics to propose meetings to partner agents, based on knowledge acquired during past interactions. There is no concept of time windows (meetings happen in exactly one time slot), and there is no negotiation as we view it.

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 [4] 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. In the final stage of such an approach, where all the constraints are known, the problem becomes one of distributed constraint satisfaction. [9] reviews a number of algorithms for such an application.

Acknowledgements

This work was supported by "Progetto Giovani Ricercatori" of the University of Bologna, and by the SOCS project, funded by the CEC, contract IST-2001-32530. The information provided is the sole responsibility of the authors and does not reflect the Community's opinion. The Community is not responsible for any use that might be made of data appearing in this publication.

References

[1] S. Conry, K. Kuwabara, V. Lesser, and R. Meyer. Multistage negotiation in distributed constraint satisfaction. *IEEE Transactions on Systems, Man and Cybernetics*, 21(6):1462–1477, November 1992.

[2] H. H. El Sakkout and M. G. Wallace. Probe backtrack search for minimal perturbation in dynamic scheduling. *Constraints. Special Issue on Industrial Constraint-Directed Scheduling*, 5(4), 2000.

[3] C. A. Harrison. *Distributed Optimisation in a Brokered Architecture.* PhD thesis, Imperial College, London, IC-PARC, William Penney Laboratory. Submitted in 2000. Available at: `http://www.icparc.ic.ac.uk/~cah1/`.

[4] S. Kraus, J. Wilkenfeld, and G. Zlotkin. Multiagent negotiation under time constraints. *Artificial Intelligence*, 75(2):297–345, 1995.

[5] F. Sadri, F. Toni, and P. Torroni. Logic agents, dialogues and negotiation: an abductive approach. In *Proceedings AISB'01 Convention, York, UK*, 2001.

[6] F. Sadri, F. Toni, and P. Torroni. An abductive logic programming architecture for negotiating agents. In *Proc. JELIA*, *LNCS* 2424, pages 419–431. Springer Verlag, 2002.

[7] F. Sadri, F. Toni, and P. Torroni. Dialogues for negotiation: agent varieties and dialogue sequences. In *Intelligent Agents VIII*, *LNAI* 2333, pages 405–421. Springer Verlag, 2002.

[8] R. Wallace. Representing possibilities in relation to constraints and agents. In *Proc. JELIA*, *LNCS* 2424, pages 407–418. Springer Verlag, 2002.

[9] M. Yokoo and K. Hirayama. Algorithms for distributed constraint satisfaction: A review. *Autonomous Agents and Multi-Agent Systems*, 3(2):185–207, 2000.

Appendix A: Definition of the predicates representing the agent states w.r.t. resources

In this Appendix, we give the semantics of the predicates representing the agent states with respect to resources, for a generic agent X. R represent resources, T_i time points between 0 and T_{max}, (Ts, Te) represent time intervals and A represent activities.

$$\begin{array}{l}
\mathcal{K}_{X,\tau} \ \vdash \ \mathbf{have}(\mathbf{R}, (\mathbf{Ts}, \mathbf{Te}), \tau) \leftrightarrow \\
\quad (\ Te \ \mathit{is} \ Ts + 1 \wedge \mathcal{K}_{X,\tau} \ \vdash \ \mathit{have}(R, Ts)\ ) \vee \\
\quad (\ Tn \ \mathit{is} \ Ts + 1 \wedge Tn < Te \wedge \mathcal{K}_{X,\tau} \ \vdash \ \mathit{have}(R, (Tn, Te), \tau) \wedge \\
\quad\ \ \mathcal{K}_{X,\tau} \ \vdash \ \mathit{have}(R, Ts)\ ) \\
\mathcal{K}_{X,\tau} \ \vdash \ \mathbf{have}(\mathbf{R}, \mathbf{T}) \leftrightarrow \\
\quad (\ \mathcal{K}_{X,\tau} \ \vdash \ \mathit{have_initially}(R, T) \wedge \\
\quad\ \ \nexists\ \tau' \leq \tau \mid \mathcal{K}_{X,\tau'} \ \vdash \ \mathit{give_away}(R, (Ta_{giv}, Tb_{giv}))) \wedge \\
\quad\quad Ta_{giv} \leq T \wedge Tb_{giv} > T\ ) \vee \\
\quad (\ \exists\ \tau' \leq \tau \mid \mathcal{K}_{X,\tau'} \ \vdash \ \mathit{obtain}(R, (Ta_{obt}, Tb_{obt}))) \wedge \\
\quad\quad Ta_{obt} \leq T \wedge Tb_{obt} > T \wedge \nexists\ \tau'' \leq \tau \mid \\
\quad\quad\ \ \mathcal{K}_{X,\tau''} \ \vdash \ \mathit{give_away}(R, (Ta_{giv}, Tb_{giv}))) \wedge \\
\quad\quad\ \ Ta_{giv} \leq T \wedge Tb_{giv} > T \wedge \tau' \leq \tau''\ ) \\
\mathcal{K}_{X,\tau} \ \vdash \ \mathbf{have_initially}(\mathbf{R}, \mathbf{T}) \leftrightarrow \\
\quad \exists\ Ts, Te \mid \mathit{have}(R, (Ts, Te), \tau)\ \in \mathcal{R}_X \wedge Ts \leq T \wedge Te > T \\
\mathcal{K}_{X,\tau} \ \vdash \ \mathbf{give_away}(R, (Ta_{giv}, Tb_{giv}))) \leftrightarrow \\
\quad \exists\ \mathit{tell}(X, _, \mathit{Subject}, _, \tau) \in \mathcal{D}_{X,\tau} \mid \\
\quad \mathit{Subject} \in \{\mathit{accept}(\mathit{request}(\mathit{give}(R, (Ta_{obt}, Tb_{obt})))), \\
\quad \mathit{accept}(\mathit{promise}(R, (Ta_{obt}, Tb_{obt}), (_, _)))\} \vee
\end{array}$$

$\exists\ tell(\_, X, Subject, \_, \tau) \in \mathcal{D}_{X,\tau} \mid$
$Subject \in \{accept(promise(R, (\_,\_), (Ta_{obt}, Tb_{obt})))\}$

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

In the following, by $\hat{\mathcal{I}}_X$ we indicate a fixed schedule of activities, for agents that negotiate at $Stage1$. We assume that both $\mathcal{I}_X$ and $\hat{\mathcal{I}}_X$ have been generated consistently (in particular, $\exists < A, R, D, Ts, Te > \in \mathcal{I}_X \leftrightarrow \exists < A, R, D, Ts', Te' > \in \hat{\mathcal{I}}_X \mid Ts' \geq Ts \wedge Te' \leq Ts \wedge Te' = Ts' + D$)

feasible$(A, (Ts, Te))$ means that a certain activity A requiring a resource R for D time units is feasible, between Ts and $Te = Ts + D$, an $need(R, (Ts, Te))$ means that the agent needs to retain R from Ts to Te.

$\mathcal{K}_{X,\tau} \ \vdash\ \mathbf{feasible1}(A, (Ts, Te)) \leftrightarrow$
$\exists\ < A, R, \_, Ts, Te > \in \hat{\mathcal{I}}_X \mid \mathcal{K}_{X,\tau} \ \vdash\ have(R, (Ts, Te), \tau)$

$\mathcal{K}_{X,\tau} \ \vdash\ \mathbf{feasible2}(A, (Ts, Te)) \leftrightarrow$
$\exists\ < A, R, D, Ts', Te' > \in \mathcal{I}_X \mid Ts' \leq Ts \wedge Te' \geq Te \wedge$
$Te - Ts = D \wedge \mathcal{K}_{X,\tau} \ \vdash\ have(R, (Ts, Te), \tau)$

$\mathcal{K}_{X,\tau} \ \vdash\ \mathbf{need1}(R, (Ts, Te), \tau) \leftrightarrow$
$\exists\ < A, R, \_, \_, \_ > \in \mathcal{I}_X \mid \mathcal{K}_{X,\tau} \ \vdash\ need1(R, (Ts, Te), A)$

$\mathcal{K}_{X,\tau} \ \vdash\ \mathbf{need1}(R, (Ts, Te), A) \leftrightarrow$
$\mathcal{K}_{X,\tau} \ \vdash\ have(R, (Ts, Te), \tau) \wedge feasible1(A, (Ts', Te')) \wedge$
$(\ Ts' \leq Ts < Te' \vee Ts' < Te \leq Te'\ )$

$\mathcal{K}_{X,\tau} \ \vdash\ \mathbf{need2}(R, (Ts, Te), \tau) \leftrightarrow$
$\exists\ < A, R, \_, \_, \_ > \in \mathcal{I}_X \mid \mathcal{K}_{X,\tau} \ \vdash\ need2(R, (Ts, Te), A)$

$\mathcal{K}_{X,\tau} \ \vdash\ \mathbf{need2}(R, (Ts, Te), A) \leftrightarrow$
$\mathcal{K}_{X,\tau} \ \vdash\ have(R, (Ts, Te), \tau) \wedge$
$\exists\ < A, R, D, Ts', Te' > \in \mathcal{I}_X \wedge Ts \geq Ts' \wedge Te \leq Te' \wedge$
$\nexists\ Ts'' \ \in [Ts', Te' - D] \mid$
$\mathcal{K}_{X,\tau} \ \vdash\ have(R, (Ts'', Ts'' + D), \tau) \wedge$
$(\ Ts'' \geq Te \vee Ts'' + D \leq Ts\ )$

We conclude with the definition of *miss*, *indifferent*, and *available1*/2.

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

$\mathcal{K}_{X,\tau} \ \vdash\ \mathbf{miss}(R, (Ts, Te), A) \leftrightarrow$
$\exists\ < A, R, \_, \_, \_ > \in \mathcal{I}_X \mid \mathcal{K}_{X,\tau} \ \nvdash\ feasible2(A, (\_,\_)) \wedge$
$\mathcal{K}_{X,\tau} \ \cup\ \{obtain(R, (Ts, Te))\} \ \vdash\ feasible2(A, (\_,\_)) \wedge$
$\nexists\ (Ts', Te') \subset (Ts, Te) \mid$
$\mathcal{K}_{X,\tau} \ \cup\ \{obtain(R, (Ts', Te'))\} \ \vdash\ feasible2(A, (\_,\_))$

$$
\begin{aligned}
&\mathcal{K}_{X,\tau} \ \vdash \ \textbf{indifferent}(R,(Ts,Te)) \leftrightarrow \\
&\qquad \mathcal{K}_{X,\tau} \ \nvdash \ \mathit{have}(R,(Ts,Te),\tau) \wedge \\
&\qquad \forall (Ts_j,Te_j) \mid \mathcal{K}_{X,\tau} \ \vdash \ \mathit{miss}(R,(Ts_j,Te_j)) \rightarrow \\
&\qquad\quad Te \leq Ts_j \vee Ts \geq Te_j \\
&\mathcal{K}_{X,\tau} \ \vdash \ \textbf{available1}(R,(Ts,Te)) \leftrightarrow \\
&\qquad \mathcal{K}_{X,\tau} \ \vdash \ \mathit{have}(R,(Ts,Te),\tau) \wedge \mathcal{K}_{X,\tau} \ \nvdash \ \mathit{need1}(R,(Ts,Te),\tau) \\
&\mathcal{K}_{X,\tau} \ \vdash \ \textbf{available2}(R,(Ts,Te)) \leftrightarrow \\
&\qquad \mathcal{K}_{X,\tau} \ \vdash \ \mathit{have}(R,(Ts,Te),\tau) \wedge \mathcal{K}_{X,\tau} \ \nvdash \ \mathit{need2}(R,(Ts,Te),\tau)
\end{aligned}
$$