A Coordination Algorithm for Multi-Agent Planning

Amal El Fallah Seghrouchni[1] and Serge Haddad[2]

[1] LIPN - URA 1507, Inst. Galilée, Université Paris-Nord
Av. J.B. Clément, 93430 Villetaneuse, France
elfallah@ura1507.univ-paris13.fr
[2] LAMSADE, Université Paris-Dauphine,
Centre Informatique, Place du Maréchal de Lattre de Tassigny
75775 Paris Cédex 16, France
haddad@lamsade.dauphine.fr

Abstract. One of the major interests of Multi-Agent Systems (MAS), which are able to handle distributed planning, is coordination. This coordination requires both an adequate plan representation and efficient interacting methods between agents. Interactions are based on information exchange (e.g. data, partial or global plan) and allow agents to update their own plans by including the exchanged information. Coordination generally produces two effects: it cancels negative interactions (e.g. resource sharing) and it takes advantage of helpful ones (e.g. handling redundant actions). A coordination model should satisfy the following requirements: domain independence, broad covering of interacting situations, operational coordination semantics and natural expression for the designer. This paper presents an adequate framework for the representation and handling of plans in MAS. It then shows how an approach based on a plan representation by means of a partial order model enables the definition of a coordination algorithm for the possible enrichment of plans.

Keywords: Multi-Agent Systems, Distributed Planning, Coordination, Plan Interactions.

1 Introduction

One of the major interests of Multi-Agent Systems (MAS) is their ability to handle distributed planning by coordinating agents' plans. Such a coordination requires both an adequate plan representation and an efficient interaction between agents. Based on information exchange (e.g. data, plans), the interaction allows agents to update their own plans by considering the exchanged information. Coordination generally produces two effects: cancelling negative interactions (e.g. harmful actions) and taking advantage of helpful interactions (e.g. handling redundant actions). Agents organize their activities and update their plans in order to cooperate and avoid conflicts. The main requirements of coordination can be summarized as follows:

- communication between agents,
- recognition of potential interactions between plans,
- negotiation between agents in the case of conflictual situations.

These aspects have already been studied in numerous papers. In coordination, the use of communication is a part of planning and action. It concerns the development of MAS where speech acts are often involved [15, 16, 3, 4, 9]. The second aspect comes from the study of interactions which have been of continuing interest in multi-agent planning [5, 8, 17, 11, 14, 7]. This mainly focuses on how planning agents can positively cooperate in distributed environments. The last aspect has been studied in [17, 6, 2]. The main criticisms to be made about most of the planning models proposed in Multi-Agent research are:

- The formal cooperation models are theoretical. Although they can be used to prove cooperation theories mathematically and show how assumptions about their domains and characteristics affect their capabilities, these models are often remote from practical systems and provide only a little help in designing MAS.
- The computing methods used to work out dependencies between agents (e.g. between actions, plans, etc.) are often static when they need to be dynamic. Even when they are dynamic, these methods often require a synchronous rhythm between execution and planification whereas the dependencies need to be handled asynchronously (i.e. we want to cover cases where one agent is building a plan while another is executing his own plan).
- The coordination process is generally centralized where it should be distributed and implies two agents where it should imply n agents. (i.e. not just two by two).

This paper focuses on the situation where n agents carry out their already coordinated plans while a new agent produces a new plan which has to be coordinated with the existing ones. A coordinating algorithm is proposed, with solutions to the representation and management of plans in view of distributed planning. The main advantages offered by the algorithm are as follows:

- the algorithm is distributed on each agent, with the result that the coordination is really distributed (i.e. each agent contributes to the coordination process),
- during coordination, no agent is suspended (i.e. each agent pursues his plan without interruption),
- the existing agents do not regenerate their plans (i.e. an agent plans once and once only) since the proposed algorithm allows coordination through the enrichment of plans,
- the algorithm solves the negative and positive interactions,
- the algorithm ensures both concurrent planning and the tolerance of possible agent failure,
- an essential feature of this algorithm is that the coordination is performed without generating all the possible plan overlaps, thus avoiding a combinatorial explosion.

Section two introduces a conceptual framework for plan management based on a classical planification formalism. Section three shows how this framework is adapted to handle a specific planification domain. Section four presents the coordination requirements, describes the specific contribution of this efficient new algorithm which can handle negative and positive interactions by coordinating plans, and gives its proof. Section five is a discussion of possible ways of improving and extending the algorithm.

2 Conceptual Framework for Plan Management

The model used here is made up of cognitive agents which represent the active components of the MAS and which evolve in the environment. The formalism is based on a classical representation often used in planning. Let us now define the framework for the plan management.

2.1 Definition of the environment

The positive or negative interactions depend on the environment, the representation of which includes the modeling of resources. The environment can be completely defined through a set of predicates called $Prop$ such that: $Prop = \{p, \bar{p}, q, \bar{q}, \cdots\}$

Notation: Let p be a positive proposition and $\bar{p}$ the negative one. A Greek letter is used for each item of $Prop$ such as: ξ, ψ, φ, etc. The negation can be extended to any sub-set of $Prop$ in the following way:
if $S \in 2^{Prop}$ then $\bar{S} = \{\bar{\xi}/\xi \in S\}$
Hence, an environment state is defined as a consistent sub-set of $Prop$, a partial state and a global state being distinguished as follows:

Definition 1. S is a partial state if and only if:
$\forall \xi \in Prop, \mid \{\xi, \bar{\xi}\} \cap S \mid \leq 1$
(i.e. an execution context which is partially defined)

Definition 2. S is a total state if and only if:
$\forall \xi \in Prop, \mid \{\xi, \bar{\xi}\} \cap S \mid = 1$

2.2 Definition of actions

The resolution plan involves actions associated with tasks to be performed. These actions constitute the plan generated by the planner. An action is defined through three components:

  - the label or the name of the action,

- the pre-conditions (Pre), which represent a partial state by means of the set of conditions which are necessary (i.e. must be satisfied) to perform the action,
- the post-conditions ($Post$), which are the set of conditions that will be satisfied after the action has been performed (add list and delete list in STRIPS representation, for instance).

Note that these conditions are syntactic constructions expressing some constraints to be satisfied. This is similar to the STRIPS rule.

Definition 3. Let $Act = \{a, b, c, \cdots\}$ be the set of possible actions, and let $a \in Act$, then:
$a \rightarrow Pre(a)$ where $Pre(a)$ is a partial state
$a \rightarrow Post(a)$ where $Post(a)$ is a partial state
The execution of the action a, when the state is S, produces the new state S' such that:

$$S \xrightarrow{a} S' \Leftrightarrow Pre(a) \subset S \text{ and } S' = (S \setminus \overline{Post}(a)) \cup Post(a).$$

2.3 Definition of plans

To reach a given goal, an agent uses a planner to elaborate a local plan. Such a plan organizes a collection of actions which can be performed sequentially or concurrently. Each agent is responsible for generating his local plans and maintaining their consistency. A correct plan execution often requires that actions be taken in some specific partial order. Whatever the total order that extends the partial order obtained from the graph, the plan must remain feasible.

Notation: Let R be an ordering relation, we note R^+ the transitive closing of R and R^* the reflexive and transitive closing of R.

Definition 4. A plan $\Pi = \{A_\pi, R_\pi\}$ is a directed acyclic graph where:
$A_\pi \subset Act$ is the set of nodes,
R_π is the set of arcs such that $R_\pi \subset A_\pi \times A_\pi$

Definition 5. Let π be a plan and $a \in A_\pi$. The necessary post-conditions $PostNec$ of an action a are defined as follows:
$PostNec(a) = \{\xi/\xi \in Post(a), \exists b \text{ such that } aR_\pi^+ b \text{ with } \xi \in Pre(b), \forall c \text{ such that } aR_\pi^+ c \text{ and } cR_\pi^+ b, \xi \notin Post(c)\}$

This definition means that the post-condition ξ which is obtained by the performance of action a is necessary for the performance of action b and that b cannot be sure of obtaining the proposition ξ otherwise. In other words, the $PostNec$ conditions are necessary for the future execution. In order to execute a plan, such post-conditions cannot be deleted. It is important to distinguish such propositions during the coordination process.
Let us note that the $PostNec$ set can be obtained automatically by a planner as

described in [15] using algorithms such as POCL [12] or TOPI [13]. The $PostNec$ are equivalent to the causal links in [1]. If another planner is used to compute the $PostNec$ set, then the cost may be high.

3   Planning Performance

3.1   Definition of a feasible plan

A plan Π is feasible if and only if each action a of Π can be performed whatever the total order that extends the partial order obtained from the graph.
An action a of Π can be performed if each pre-condition ξ of a can be satisfied. This is possible if there exists a strict predecessor b of a which generates the pre-condition ξ, and for all action c which generates $\bar{\xi}$, then c must be a predecessor of b or a successor of a.

Definition 6. A plan Π is feasible if and only if:
$\forall a \in A_\pi$, $\forall \xi \in Pre(a)$, $\exists b \in A_\pi$, such that:

- $\xi \in Post(b)$
- $bR_\pi^+ a$ (strict predecessor)
- $\forall c \in A_\pi$ such that $\bar{\xi} \in Post(c)$ $cR_\pi^* b$ or $aR_\pi^* c$

Hypothesis: We assume that each agent has a set of stable states where an agent is idle. Such states are called homestates and can be defined as a family of states to which an agent can come back and from which he can easily generate a plan. In order to build a structured plan, we also assume that an agent proceeds in two phases.

Phase 1. The agent builds his plan independently of the initial state. An agent starts from the current state and tries to come back to a family of homestates.

Phase 2. The agent tries to build a structured plan from the first phase.

3.2   Definition of a structured plan

Definition 7. A plan $\Pi = \{A_\pi, R_\pi\}$ is a structured plan if and only if:

- $\forall a_i \in Init_\pi$, $\forall a \in A_\pi$ we have not $aR_\pi a_i$
- $\forall a \in A_\pi$, $\exists a_i \in Init_\pi$ such that $a_iR_\pi^* a$
- $\forall a \in A_\pi$, $\exists a_e \in End_\pi$ such that $aR_\pi^* a_e$

where:
$$A_\pi = Init_\pi \biguplus Int_\pi \biguplus End_\pi$$
and $\biguplus$ is the disjoint union of:

- $Init_\pi$: the set of initial actions in the plan Π generated by the first phase,
- Int_π: the set of intermediate actions in Π through which the intermediate results are obtained,
- End_π: the set of the end actions which give the final results of the performed plan.

These constraints mean that no initial action can have a predecessor, and each intermediate action has both an end action as a successor and an initial action as a predecessor.

4   Coordination Requirements

In order to introduce the coordination process, let us consider an MAS with n agents $\{g_1, g_2, \ldots, g_n\}$, to each of which is associated a plan: $g_i \to \Pi_i$. Let us also assume that the plans $\Pi_1, \Pi_2, \ldots, \Pi_{n-1}$, (associated respectively with $g_1, g_2, \ldots, g_{n-1}$) are already coordinated and that the agent g_n produces a new plan Π_n. The problem to be solved here is to coordinate the plan Π_n by enriching it without modifying the existing plans $\Pi_1, \Pi_2, \ldots, \Pi_{n-1}$. Moreover, the coordination process must generate both a feasible plan for g_n and a solution for possible positive and negative interactions.

4.1   The coordination algorithm involving n agents

The coordination algorithm (COA) involves all the agents during the two phases: the first one is a general coordination preparation, the results of which are applied dynamically in the second phase.

Phase 1.   1. $g_1, g_2, \ldots, g_{n-1}$ perform respectively the plans $\Pi_1, \Pi_2, \ldots, \Pi_{n-1}$ which are already coordinated. The agent g_n produces a new plan.

2. g_n sends his plan Π_n to the existing agents.

3. Each agent $g_k (1 \leq k \leq n-1)$ receives Π_n and tries to coordinate it with his own plan Π_k.

4. $g_1, g_2, \ldots, g_{n-1}$ return to g_n the coordination results in the form of arcs representing the possible synchronization between the two plans.

The last two points represent the local coordination for each agent. Let us assume that the agent $g_i (1 \leq i \leq n-1)$ handles his own plan Π_i, and Π_n received from g_n.

To begin with, the agent g_i synchronizes Π_i and Π_n by creating an arc between each pair of actions in $End_{\Pi_i} \times Init_{\Pi_n}$. Each $arc(a_j^i, a_k^n)$ has an action a_j in End_{Π_i} as a source and an action a_k in $Init_{\Pi_n}$ as a target (see Situation 1
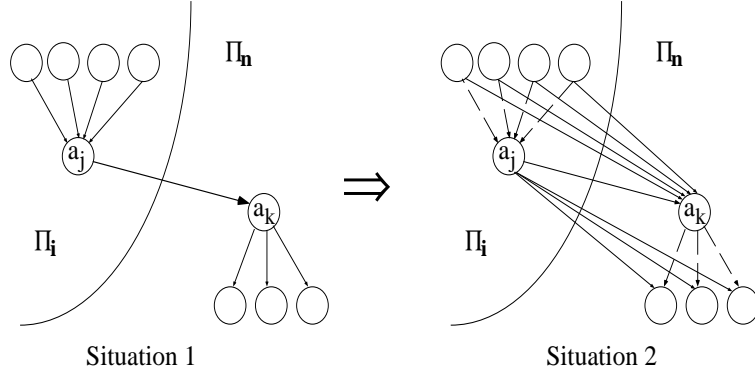
Fig. 1. Moving up arcs.

in Figure. 1). Note that the action index represents the action index in the plan while the action exponent represents the plan index in the system.

Then the agent g_i starts from the synchronization arcs and tries to move up through the arcs in order to constitute the set of possible interactions as follows. While there exists a synchronization arc which verifies the condition

$Arc = \{arc(a_j^i, a_k^n)$ such that:

- $1 \leq i \leq n-1$ and $1 \leq j \leq |A_i|$ and $1 \leq k \leq |A_n|$,
- $\overline{Post}(a_j^i) \cap PostNec(a_k^n) = \emptyset$,
- $\overline{Pre}(a_j^i) \cap Post(a_k^n) = \emptyset\}$.

move up through the arcs (see Situation 2 in Figure. 1) such that:
$Suc(a_j) = Suc(a_k)$ and the $Pred(a_k) = Pred(a_j)$
where:
$Pred(a) = \{b \in A_{\Pi_i}$ such that $bR_{\Pi_i}a\}$
$Suc(a) = \{c \in A_{\Pi_n}$ such that $aR_{\Pi_i}c\}$

Phase 2. The agent g_n starts performing his plan Π_n by taking into account the synchronization arcs (i.e. $arc(a_j^i, a_k^n)$). The coordinated plan Π_n' is:

$\Pi_n' = \Pi_n \cup arc(a_j^i, a_k^n)$
where
$(1 \leq i \leq n-1)(1 \leq j \leq |A_i|)$ and $(1 \leq k \leq |A_n|)$

When the action a_j^i is performed the agent g_i sends synchronization messages to g_n indicating that the arcs (a_j^i, a_k^n) are valid. Consequently, an action a_k^n depending on arcs (a_j^i, a_k^n) will be executed by g_n only if all the corresponding messages are received.
Note that if all the $PostNec(a_k^n)$ of an action have already been generated

by the existing agents, the agent g_n does not execute a_k^n. In the following, we will note $Q_{i,j}$ the set of the synchronisation arcs which are created from the plan Π_i to the plan Π_j such that:

$Q_{i,j} = arc(a_k^i, a_l^j)$ where $1 \leq i, j \leq n$, $1 \leq k \leq \mid A_i \mid$, and $1 \leq l \leq \mid A_j \mid$

Let us now prove that global performance remains possible. In other words, we must prove that the union of all the plans and the new arcs $Q_{i,j}$ is a feasible plan.

Theorem 8. *The union of the plans (i.e. existing ones and the new plan) and the new sets of $Q_{i,j}$ generated by the algorithm COA:*

$$\left(\bigcup_{i=1}^{n} \Pi_i\right) \bigcup \left(\bigcup_{i=1}^{n} \left[\bigcup_{j=i+1}^{n} Q_{i,j}\right]\right)$$

is a feasible plan.

4.2   Proof of the coordination algorithm

Notation:
    Let $\Pi^{(1)}, \cdots, \Pi^{(n)}$ be the coordinated plans of the agents $\{1\}$, $\{1, 2\}$, ..., $\{1, \ldots, n\}$. We prove by recurrence that the plan $\Pi^{(n)}$ is feasible. First, we introduce new notations:

- $\Pi_{\{i,j\}} = \Pi_i \cup \Pi_j \cup Q_{i,j}$ *if $i < j$*
- $\Pi_{\{i,i\}} = \Pi_i$

Our recurrence hypothesis for a plan $\Pi^{(i)}$ is:
    $\forall a \in A_{\Pi_j}$, $\forall \xi \in Pre(a), \exists b \in A_{\Pi_j}$ such that:

- $\xi \in PostNec(b)$
- $b R_{\Pi_j}^+ a$
- $\forall c \in A_{\Pi_k}$ such that $\bar{\xi} \in Post(c)$ then

$$c R_{\Pi_{\{j,k\}}}^* b \ or \ a R_{\Pi_{\{j,k\}}}^* c$$

Initially, $\Pi^{(1)} = \Pi_1$ is feasible and satisfies the recurrence hypothesis because all the actions belonging to the agent (1) are coordinated. Let us assume that $\Pi^{(n-1)}$ satisfies the recurrence hypothesis and let us prove that $\Pi^{(n)}$ also satisfies it. We can note a series of plans $\Pi^{(n)(1)}, \cdots, \Pi^{(n)(s)}$ where:

- $\Pi^{(n)(1)}$ is the plan which is obtained from $\Pi^{(n-1)}$ by adding Π_n and the initial synchronization arcs $Q_{i,n}$ as follows:

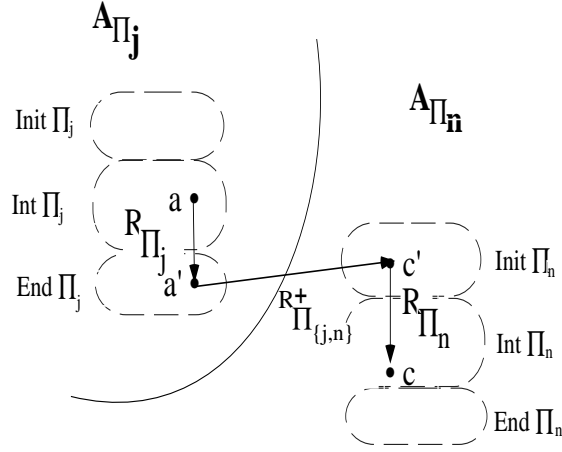$$\Pi^{(n)(1)} = \Pi^{(n-1)} \cup \Pi_n \bigcup_{i<n} Q_{i,n}$$

Fig. 2. A structured plan

- $\Pi^{(n)(m+1)}$ is the plan obtained from $\Pi^{(n)(m)}$ by substituting other arcs for the synchronization arc $\alpha \to \beta$ with $\alpha \in \Pi^{(i)}$ and $\beta \in \Pi^{(n)}$ during examination of the plan.

First, we prove that $\Pi^{(n)(1)}$ satisfies the recurrence hypothesis:
$\forall a \in A_{\Pi_j}, \ \forall \xi \in Pre(a)$

case 1: <u>If $j \neq n$</u> then the recurrence hypothesis implies:

$\exists b \in A_{\Pi_j}$ such that:
- $\xi \in PostNec(b)$
- $b R^{+}_{\Pi_j} a$
- $\forall c \in A_{\Pi_k} (k \neq n)$ such that $\bar{\xi} \in Post(c)$ then
$$c R^{*}_{\Pi_{\{j,k\}}} b \ or \ a R^{*}_{\Pi_{\{j,k\}}} c$$

Let us now benefit from our structured plan definition (see definition 7 and Figure. 2) and let $c \in A_{\Pi_n}$, then:
- $\exists c' \in Init_{\Pi_n} : c' R^{*}_{\Pi_n} c$
- $\exists a' \in End_{\Pi_j} : a R^{*}_{\Pi_j} a'$
- $\exists (a', c') \in Q_{\{j,n\}}$
hence $a R^{*}_{\Pi_{j,n}} c$

case 2: <u>If $j = n$</u> then, since Π_n is a feasible plan, $\exists b \in A_{\Pi_n}$ such that:

- $\xi \in PostNec(b)$
- $b R^{+}_{\Pi_n} a$
- $\forall c \in A_{\Pi_n}$ such that $\bar{\xi} \in Post(c)$ then

$$cR^*_{\Pi_n}b \ or \ aR^*_{\Pi_n}c$$

Let $c \in A_{\Pi_k}(k \neq n)$, then:
- $\exists c' \in End_{\Pi_k} : cR^*_{\Pi_k}c'$
- $\exists a' \in Init_{\Pi_n} : a'R^*_{\Pi_{\{k,n\}}}b$
- $\exists(c',a') \in Q_{k,n}$

hence $cR^*_{\Pi_{\{k,n\}}}b$

We conclude then that $\Pi^{(n)(1)}$ satisfies the recurrence hypothesis. All that remains to be down is to prove that if $\Pi^{(n)(m)}$ satisfies the recurrence hypothesis then $\Pi^{(n)(m+1)}$ also satisfies it. Let us call $\alpha \rightarrow \beta$ (with $\alpha \in \Pi^{(i)}$ and $\beta \in \Pi^{(n)}$) the substituted arc.

$\forall a \in A_{\Pi_j}$, $\forall \xi \in Pre(a)$, the recurrence hypothesis implies that $\exists b \in A_{\Pi_j}$ such that:

- $\xi \in PostNec(b)$
- $bR^+_{\Pi_j}a$

Let us assume that $\exists c \in A_{\Pi_k}$ such that $\bar{\xi} \in Post(c)$. Let us now consider all the possible cases:

case 1: ($j \neq i$ and $j \neq n$) or ($k \neq i$ and $k \neq n$)

As $\Pi_{\{j,k\}}$ is unchanged by the substitution then $cR^*_{\Pi_{\{j,k\}}}b$ or $aR^*_{\Pi_{\{j,k\}}}c$

case 2: ($j = i$ and $k = i$)

As Π_j is unchanged by the substitution then $cR^*_{\Pi_{\{j,k\}}}b$ or $aR^*_{\Pi_{\{j,k\}}}c$

case 3: ($j = n$ and $k = n$)

As Π_n is unchanged by the substitution then $cR^*_{\Pi_{\{j,k\}}}b$ or $aR^*_{\Pi_{\{j,k\}}}c$

case 4: ($j = n$ and $k = i$)

In $\Pi^{(n)(m)}$ we have $cR^*_{\Pi_{\{j,k\}}}b$ or $aR^*_{\Pi_{\{j,k\}}}c$

since $j = n$ then necessarily $cR^*_{\Pi_{\{j,k\}}}b$ (because in $Q_{i,n}$ the arcs are oriented from i to n).

Let (δ, γ) be the only arc on the path from c to b where $\delta \in \Pi_i$ and $\gamma \in \Pi_n$. The situation in $\Pi^{(n)(m)}$ is then:

case 4.a: $(\delta, \gamma) \neq (\alpha, \beta)$

The situation is unchanged in $\Pi^{(n)(m+1)}$ (see Figure. 3).

case 4.b: $(\delta, \gamma) = (\alpha, \beta)$

case 4.b.1: If $(c \neq \delta)$ then $\exists c'$ such that $cR^*_{\Pi_i}c'$ and $c'R_{\Pi_i}\delta$ then $cR^*_{\Pi_{\{j,k\}}}b$ (see Situation 1 in Figure. 4)
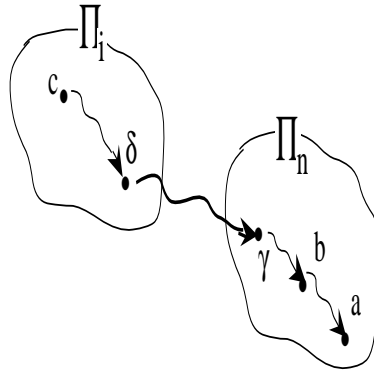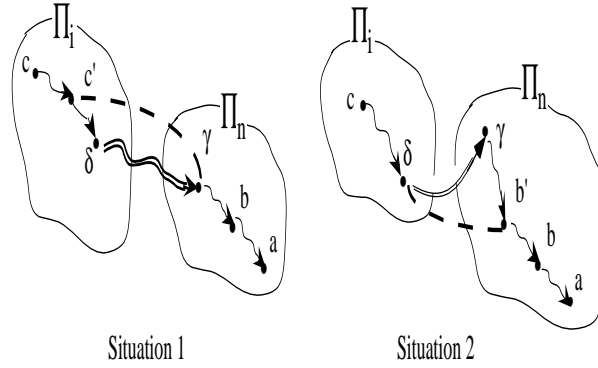
Fig. 3. Unchanged arc

Situation 1   Situation 2

Fig. 4. Substituted arcs

case 4.b.2: If $(b \neq \gamma)$ then $\exists b'$ such that $b' R^*_{\Pi_n} b$ and $\gamma R_{\Pi_n} b'$ then the situation in $\Pi^{(n)(m+1)}$ becomes (see Situation 2 in Figure. 4): $c R^*_{\Pi_{\{j,k\}}} b$

case 4.b.3: If $(c = \delta = \alpha \; and \; b = \gamma = \beta)$ then $\xi \in PostNec(\beta) \; and \; \bar{\xi} \in Post(\alpha)$ which is an impossible case.

case 5: If $(j = n$ and $k = n)$ then in $\Pi^{(n)(m)}$ we have $c R^*_{\Pi_{\{j,k\}}} b$ or $a R^*_{\Pi_{\{j,k\}}} c$ since $j = n$ then necessarily $a R^*_{\Pi_{\{j,k\}}} c$ (as in case 4).

Let (δ, γ) be an arc as defined before (see case 4), the situation in $\Pi^{(n)(m)}$ (see Figure. 5) presents two sub-cases:

case 5.a: if $(\delta, \gamma) \neq (\alpha, \beta)$ then the $\Pi^{(n)(m+1)}$ situation is unchanged.

case 5.b: if $(\delta, \gamma) = (\alpha, \beta)$ then

    5.b.1: $a \neq \delta$ (as in case 4)
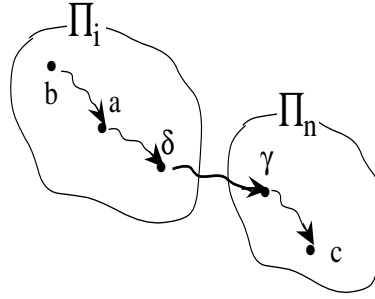
Fig. 5. Unchanged or impossible case

5.b.2: $c \neq \delta$ (as in case 4)

5.b.3: if $(a = \delta = \alpha$ and $c = \gamma = \beta)$
then $\xi \in Pre(\alpha)$ and $\bar{\xi} \in Post(\beta)$ which is an impossible case.

5 Discussion and conclusion

Different research has been done in the area of distributed planning and, in particular, incremental planning. Some of the approaches are similar to the one presented in this paper and include the following.

The approach suggested by V. Martial [11] offers a number of advantages in that the model is a theoretical one which handles both positive and negative interactions. It is however limited by the fact that only two agents (point to point) can be coordinated at a time whereas it would be more useful to be able to coordinate n agents, thus providing a really distributed coordination. The approach put forward by R. Alami [10] is based on the paradigm of plan merging. Although it is robust and handles n agents at a time, it is centralized by the new agent. Moreover, the existing agents are suspended during the process and the new plan is generated in function of existing ones. This is incompatible with the idea of an autonomous agent who can produce his own plan to be coordinated later. Other research focuses on distributed planning [2] but is based rather on organizational structures [18].

The approach described in this paper provides a formal framework to deal with coordinating plans in the area of distributed planning. Both plans and the environment components are defined exactly. A distributed algorithm is proposed to coordinate plans; it is based on the enrichment of plans without modifying existing ones. In this approach the planning and communication aspects are merged, which offers a number of advantages.

- It is a distributed algorithm.
- It coordinates n agents at once.

- It can cancel negative interactions (through the synchronization arcs) and take advantage of positive ones (an action is not executed systematically). In addition, if the different inter-dependencies between agents are specified in advance (predicated) they are only taken into account gradually during execution, thus offering great flexibility and allowing for agent failures.
- No replanning is necessary.
- The approach is generic, independent of the application domain and can be applied to a large range of applications.
- The formalism used is simple and easy to understand.
- The *PostNec* type conditions can be generated by any classical planner.
- The conditions resulting from the planner may cover different sorts of constraints: resource sharing, presence or absence of events, etc. The distinctions have not been drawn here since all the conditions are represented in the predicates where domains of variables can be extended or reduced depending on the domain in question.

This approach can be improved by introducing the notions of abstraction and refinement in the plan. The plan may contain different levels of abstraction used according to the nature of the operation realized on it. Abstraction is necessary to deal with abstract actions which may be refined in order to take into account the favor relations [11].

References

1. A. Barrett and D.S. Weld. Characterizing Subgoal Interactions for Planning. In proceedings of IJCAI-93, pp 1388-1393, 1993.
2. A.H. Bond and L. Gasser. Reading on DAI. Morgan Kauffman Publishers, Inc. 1988.
3. P.R. Cohen and H. Leveque. Teamwork. Tr 503. SRI International, 1991.
4. P.R. Cohen, J. Morgan, and M.E. Pollack (editors). Intentions in communication. MIT Press, 1990.
5. D.D. Corkill. Hierarchical Planning in a Distributed Environment. In Proc. of the sixth International Joint Conference on Artificial Intelligence (IJCAI-79), 1979.
6. R. Davis and R. Smith. Negotiation as a Metaphor for distributed Problem Solving. In Proc. of Artificial Intelligence, vol 20, pp 63-109, 1983.
7. E. Ephrati and J.S. Rosenschein. Constrained Intelligent Action: Planning Under the Influence of a Master Agent. In Proc. of the tenth National Conference on Artificial Intelligence (AAAI-92), 1992.
8. M.P. Georgeff. Communication and interaction in Multi-Agent Planning. In Proc. of the third National Conference on Artificial Intelligence (AAAI-83), 1983.
9. B.J. Grosz and C.L. Sidner. Attention, intentions and structure of discourse. Computational Linguistics, 12(3), 1986.
10. F. Ingrand, R. Alami and F. Robert. Multi-Robot Cooperation through Incremental Plan Merging. Submitted to International Conference on Robotics and Automation, Nagoya. Japan, 1995.

11. V. Martial. Coordination of Plans in a Multi-Agent World by Taking Advantage of the Favor Relation. In Proc. of the tenth International Workshop on Distributed Artificial Intelligence, 1990. and Coordinating plans of autonomous agents, Springer Verlag, 1991.

12. D. McAllester and D. Rosenblitt. Systematic Non-linear Planning. In proceedings of AAAI-91, pp 634-639, 1991.

13. N. Nilsson. Principles of Artificial Intelligence. Tioga Publishing company, Palo Alto, CA, 1980.

14. E. Osawa and M. Tokoro. Collaborative Plan Construction for Multi-Agent Mutual Planning. In E.Werner and Y.Demazeau, DECENTRALIZED A.I.3. Elsevier/North Holland, 1992.

15. J.R. Searle. Collective Intentionality. In Cohen et al, 1990

16. J.R. Searle. Collective intentions and actions. In: PR Cohen, J. Morgan and ME Pollac (eds.), Intentions in Communication, pp 401-416. MIT Press, 1990.

17. G. Zlotkin and J.S. Rosenschein. Negociation and Task Sharing in a Cooperative Domain. In Proceedings of the ninth Workshop on Distributed Artificial Intelligence, 1989.

18. E. Werner. Cooperating agents: a unified theory of communication and social structure. In L. Gasser and MN. Huhns (eds.), Distributed Artificial Intelligence Vol II, pp 3-36. Pitman, 1989.

This article was processed using the LaTeX macro package with LLNCS style