Multi-robot Cooperation : Architectures and Paradigms

Rachid ALAMI
LAAS-CNRS,
7, Avenue du Colonel Roche
31077 Toulouse Cedex 4, France
Rachid.Alami@laas.fr

Résumé

This paper presents a generic architecture for the operation of a team of autonomous robots to achieve complex missions. Its interest stems from its ability to provide a framework for cooperative decisional processes at different levels : high level plan synthesis, task allocation and task achievement. It is based on a combination of local individual planning and coordinated decision for incremental plan adaptation to the multi-robot context.

Indeed, we claim that it is often possible (and useful) to treat these three issues separately. As we will see, this levels deal with problems of different nature, leading to specific representations, algorithms and protocols.

Mots Clef

Multi-robot cooperation, coordination, cooperative task allocation, control architectures

1 Introduction

We propose a generic architecture for the operation of a team of autonomous robots. This architecture is based on a combination of local individual planning and coordinated decision for incremental plan adaptation to the multi-robot context. It has been designed to cover issues ranging from mission planning for several robots, to effective conflict free execution in a dynamic environment. It is aimed not only to integrate our past contributions but also to allow to investigate new cooperation and coordination schemes.

After a brief analysis of related work, we present an overview of the architecture. We will successively address (1) a distributed task allocation protocol and (2) a cooperative task achievement scheme that detects and treats resource conflict situations as well as sources of inefficiency.

The overall system allows a set of autonomous robots not only to perform their tasks in a coherent and non-conflict manner but also to cooperatively enhance their performance taking into account the robots capabilities as well as their execution context.

2 Related work

Research devoted to multi-robot systems[Dudek, 1997, Cao et al., 1997, Parker, 2000] covers a large spectrum of topics. We limit our analysis of related work to contributions proposing cooperative schemes at the architectural and/or decisional level.

In such stream, *behavior-based* and similar approaches [Mataric, 1994, Mackenzie and Arkin, 1997], propose to build sophisticated multi-robot cooperation through the combination of simple (but robust) interaction behaviors. ALLIANCE [Parker, 1998] is a distributed behavior based architecture, which uses mathematically modeled motivations that enable/inhibit behaviors, resulting in tasks (re)allocation and (re)decomposition.

AI-based cooperative systems have proposed domain independent models for agents interaction. For example, [Boutilier and Brafman, 1997] and [Ephrati et al., 1994] enrich the STRIPS formalism, aiming to build centralized/decentralized conflict-free plans, while [Clement and Durfee, 1999] develops specialized agents which are responsible for individual plans coordination.

Several generic approaches have been proposed concerning goal decomposition, task allocation and negotiation [Asama and Ozaki, 1991, DesJardins et al., 1999]. PGP [Durfee and Lesser, 1987] (and later GPGP [Decker and Lesser, 1992]) is a specialized mission representation that allows exchanges of plans among the agents. DIPART [Pollack, 1996] is a scheme for task (re)allocation based on load balancing. Cooperation has also been treated through negotiation strategies [Rosenschein and Zlotkin, 1994] like CNP-based protocols [Smith, 1980], or BDI approaches where agents interaction is based on their commitment to achieve individual/collective goals [Jennings, 1995, Sullivan et al., 1999]. Another perspective is based on the elaboration of conventions and/or rules. For instance, “social behaviors” [Shoham and Tennenholtz, 1995] have been proposed as a way to program multi-agent systems. In STEAM [Tambe, 1998], coordination rules are designed in order to facilitate the cohesion of the group and the programming of its activities.

Coordination while achieving independent goals has been mostly addressed in the framework of application-specific techniques such as multi-robot cooperative navigation [Yuta and Premvuti, 1992, Brumitt, 1996, Azarm and Schmidt, 1997]. There are also efforts to build decentralized algorithms for specific tasks like cooperative manipulation [Wang and Kumar, 2002, Gravot and Alami, 2002] or environment mapping [Burgard et al., 2002].

3 A multi-robot architecture for incremental plan enhancement

The generic architecture that we propose covers issues ranging from mission planning for several autonomous robots, to effective conflict free execution in a dynamic environment.

This architecture is based on a combination of local individual planning and coordinated decision for incremental plan adaptation to the multi-robot context. It is built on the assumption that, in a complex system composed of several autonomous robots equipped with their own sensors and effectors, the ability of a given robot, to achieve a given task in a given situation can be best computed using a planner. Indeed, we claim that the robots must be able to plan/refine their respective tasks, taking into account the other robots' plans as planning/refinement constraints, and thus producing plans containing coordinated and cooperative actions that ensure their proper execution and will serve as a basis for negotiation.

It remains to determine what are the relevant decisional problems that should be addressed. The architecture we propose is precisely an answer to this question. It provides a framework where multi-robot decisional issues can be treated at three different levels : the *decomposition* of a mission into tasks (mission planning), the *allocation* of tasks among the available robots and the *tasks achievement* in a multi-robot context (Figure 1).

Indeed, we claim that it is often possible (and useful) to treat these three issues separately. As we will see, this levels deal with problems of different nature, leading to specific representations, algorithms and protocols.

This architecture is directly derived from the LAAS[1] architecture [Alami et al., 1998a]. It involves a hierarchy of three decisional levels having different temporal constraints and manipulating different data representations. Each level has a reactive (supervisor) and a deliberative component (planner, plan-merger…).

Communication between robots can take place at a different levels. For a given level, both components communicate with their corresponding component. The reactive components exchange *signals* and run *protocols* ; the deliberative components exchange *plans*, *goals* and data.

Let us examine the three levels with more details.

3.1 Mission Planning and Supervision

This is a pure plan synthesis problem. It consists in decomposing a mission, expressed at a very high level, into a set of partially ordered tasks that can be performed by a given team of robots. One can consider that this plan elaboration process is finished when the obtained tasks have a sufficient range and are sufficiently independent to allow a substantial "selfish" robot activity.

We assume that there is no need at this level to know precisely the current robots states. It should be enough to know the types of available robots, their number, their high level features.

An example of such a mission could be transporting and assembling a superstructure in a construction site. It may require to synthesize a sophisticated plan composed of numerous partially ordered tasks to be performed by various robot types with different capabilities [Gravot et al., 2003] : transport of heavy loads, maneuvers in cluttered environment, manipulation…

Mission decomposition is a purely deliberative. It is at this level that there are less needs of context dependent information. It can be done in a central way. Indeed, it is essentially a one thread process.

Of course it can benefit from several CPUs but this is a distribution of computing load, which is different in nature from problems calling for cooperative decision-making based on independent goals, on various robot capabilities and contexts.

In our current implementation, mission planning is produced by a central high level planner, for instance IxTeT [Laborie, 1995, Lemai 04], or the mission is provided directly by the user as a set of partially ordered tasks.

3.2 Task allocation among the robots

At this level, a mission is a set of partially ordered tasks, where each task (T_i) is defined as a set of goals to be achieved. The tasks are allocated to the robots based on their capabilities and on their execution context.

This level is not necessarily distributed. However, its distribution is clearly preferred since task allocation is essentially based on proper or local information.

We have implemented this level through M+NTA[2] [Botelho and Alami, 1999]. The tasks are allocated (and re-allocated when necessary) incrementally through a negotiation process between robot candidates. This negotiation is derived from the Contract-Net Protocol [Smith, 1980]. It is combined with a task planning and cost estimation activity which allows each robot to decide its future actions taking into account its current context and task, its own capacities as well as the capacities of the other robots.

Note that multi-robot task allocation is now well explored [Gerkey and Mataric 04, Dias et al ;, 2005]. An illustrative example can be found in [Lemaire 04] where task allocation has been studied in the framework of the Comets project [Comets-url].

Role assignment can also be performed at this level. Such an activity can also be performed in a distributed

[1] LAAS : LAAS' Architecture for Autonomous Systems.

[2] NTA : NEGOTIATION FOR TASK ALLOCATION

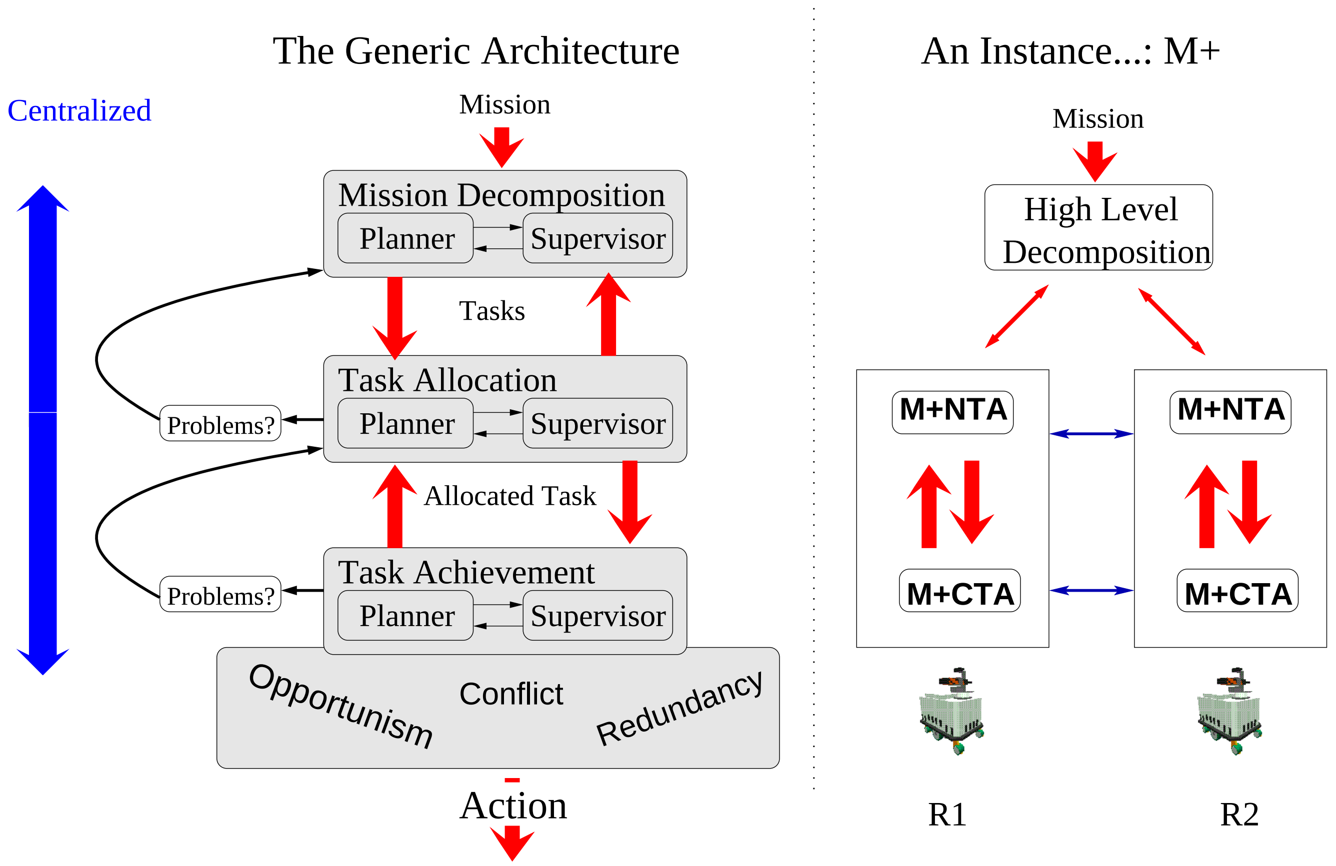

FIG. 1 – Our architecture for multi-robot cooperation. NTA stands for " Negotiation for task allocation" and CTA stands for "Cooperative Task Achievement"

way[Tambe, 1998, Gancet 05] and give the opportunity for context-based negotiation.

3.3 Task achievement in a multi-robot context

The allocated tasks, and this is a key aspect in robotics, cannot be directly "executed" but require further refinement taking into account the execution context [Alami et al., 1998a].
Since each robot synthesizes its own detailed plan, we identify two classes of problems related to the distributed nature of the system : (1) coordination to avoid and/or solve resource conflicts and (2) cooperation to enhance the efficiency of the system. The first class has been often treated in the literature. The second class is newer and raises some interesting cooperative issues linked to the improvement of the global performance by detecting sources of inefficiency and proposing possible enhancements.

Coordination to avoid conflicts

Each robot, while seeking to achieve its goal will have to compete for resources, to comply with other robots activities. Indeed, the higher levels, even if they produce valid mission decomposition, do not consider all possible conflicts that may appear at task execution level. We have already treated resource conflict situations as well as coordinated navigation [Alami et al., 1997, Gravot and Alami, 2001]. We will see, in the sequel, that the Plan-Merging Paradigm can be extended to more general conflicts.

Cooperation to enhance the system performance

We have identified several cooperative issues based on local interactions :

1. **opportunistic action re-allocation** : one robot can opportunistically detect that it will be beneficial for the global performance if it could perform an action that was originally planned by another robot ;
2. **detection and suppression of redundancy** : it may happen that various robots have planned actions which lead to the same world state. There should be some reasoning capabilities to allow them to decide when and which robot will perform actions that lead to the desired state while avoiding redundant executions ;

3. **incremental/additive actions** : the robots detect that an action originally planned by one robot can be incrementally achieved by several robots with a "cumulative" effect and that this could be beneficial to the global performance.

In our current instantiation of the architecture, M+CTA[3] implements [Botelho and Alami, 2000, Alami and Botelho, 2002] this incremental task achievement level.

3.4 Discussion on main design issues

In the following we discuss some design issues relative to our architecture. Architectural choices may often be considered somehow as arbitrary. Our design is partially intuitive and partially based on our own observations and on the main domains in the literature where multi-robot cooperation has been applied.

One, two or three levels. It may happen that for some applications, it is impossible to separate the mission decomposition and the task allocation aspects because they are too tightly linked. This is the case when the mission decomposition depends heavily not only on the types of robots available in the environment but also on their number and their current situation. In such case, the two levels should be merged in a one step planning process.
The frontier between levels that corresponds to a real qualitative change is between the task allocation and the task achievement levels. But, of course, it is still possible to devise intricate examples that challenge any architectural decomposition.
For instance, in the great majority of multi-robot control architectures described in the literature, only one aspect or the other is addressed. But this is only possible if the other aspects are simplified. At the highest level, the mission is often given already decomposed or with a small number of (trivial) decompositions. For example : transferring a bunch of n objects is trivially decomposed in n transfer tasks of individual objects. Numerous other possibilities (perhaps more efficient) may exist depending on the types of objects, the robot capabilities and their current state…
In numerous multi-mobile robot systems, elaborated motion coordination - which clearly belongs to the task achievement level - is neglected or ignored. Such simplification is acceptable only for non constrained environments where local non-coordinated obstacle avoidance schemes are sufficient.

Cooperative skills. Not all levels are activated or even present on all robots in a given application. For instance, one can imagine, in a hospital environment, the operation of several teams of mobile robots : a cleaning robots team, a meals and linen delivery team, and a set autonomous wheel-chairs (some of them do not even belong to the hospital)

[3]CTA : COOPERATIVE TASK ACHIEVEMENT

The robots within cleaning team may cooperate together at mission level. The meals and linen delivery team may cooperate at task allocation level. All robots need at least coordinate their use of common resources ; indeed, this is mandatory level.

Global coherence and efficiency. While the architecture may be considered as satisfactory in terms of identification of the relevant levels of abstractions and their articulation, this is not a guarantee of global coherence nor of efficient operation of the robots.
Indeed, such properties depend primarily on the cooperative schemes and the algorithms that are implemented *inside* each level. For example, the Plan-Merging Paradigm has been devised to provide incremental plan adaptation while maintaining two key features [Qutub et al., 1997, Alami et al., 1998b] :
– the coherence of the global scheme and the ability to detect the situations where it is not applicable
– a localized management of the planning and coordination processes with, in particularly intricate situations, a progressive transition to more global schemes which may "degrade" to a unique and centralized planning activity.
We describe, in the sequel, the Plan-Merging Paradigm and show how it can be used for distributed incremental plan adaptation.

4 Plan-Merging for Cooperative Task achievement

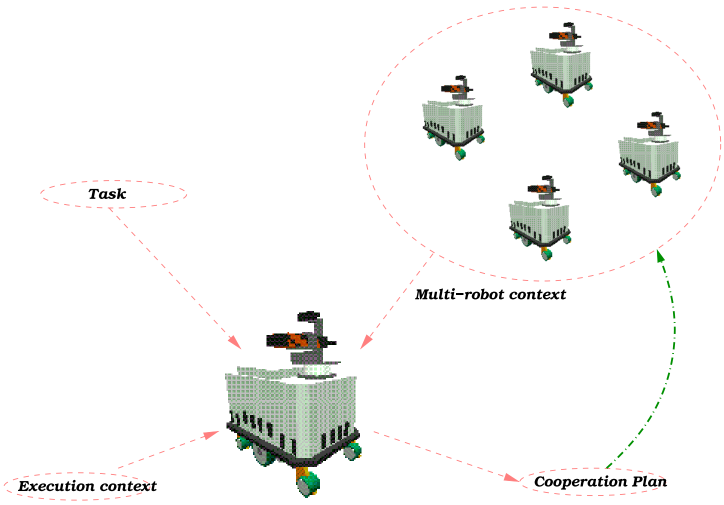

FIG. 2 – The Plan Merging Paradigm

The Plan-Merging Paradigm (Figure 2) has been initially developed in the framework of coordinated resource use in a constrained environment[Alami et al., 1995, Alami et al., 1997, Alami et al., 1998b].

The Plan-Merging scheme involves two aspects : (1) the protocol that defines control a distributed decision for plan adaptation that we have called "plan-merging" and (2) the algorithmic part of the operations on plans performed within this framework.
We will restrict ourselves, in the sequel, to the protocol aspect, i.e. the incremental adaptation of a robot plans to the multi-robot context. The interested reader may refer to [Alami et al., 1997, Gravot and Alami, 2001, Alami and Botelho, 2002] for the algorithmic issues.

4.1 The Plan-Merging Protocol

Let us assume that we have a set of autonomous robots and a higher-level system (users, a central station or a higher decisional level) which, from time to time, sends tasks to robots. Tasks are expressed as individual goals to achieve.
Whenever a robot receives a new goal, it elaborates an *Individual Plan* which takes as initial state the final state of its current plan.
An action *a* in this context has a temporal extent. It can be represented by a set of events that are partially ordered. There are two particular events : the *start* and the *end* events.
A robot plan can be represented by : $P = (I, A, E, L)$ where :
- I is the initial state
- A is a set of actions
- E is a set of events (including all actions *start* events)
- L is a set of temporal order relations between events $L = \{(e_i < e_j)\}$. The set of all temporal relations between events is a DAG (Directly Acyclic Graph).

When R_i receives its $(k+1)$-th goal G_i^{k+1}, it elaborates a plan IP_i^{k+1} which achieves it. This is an Incremental Planning step : $IP_i^{k+1} = PLAN(IP_i^k, G_i^{k+1})$
Plan updating happens when events occur : $IP_i^k = UPDATE(e, IP_i^k)$. If the next event is a *start* then the corresponding action is started.

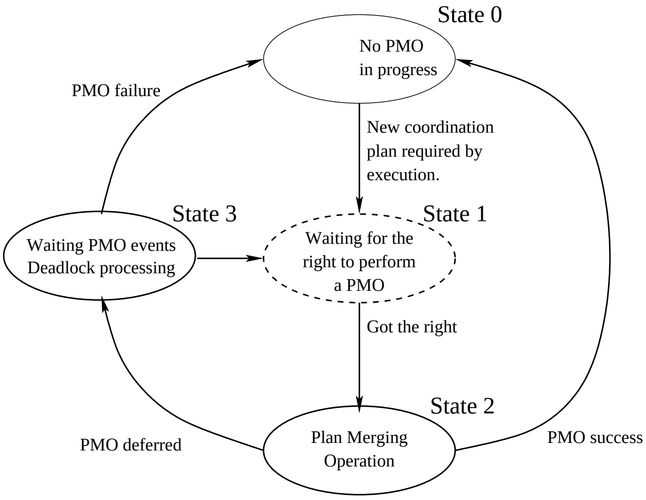

FIG. 3 – The Plan Merging Protocol.

However, in the Plan-Merging scheme, before executing this plan, the robot must ensure that it is valid in the multi-robot context. Indeed, it is at least necessary to detect and solve all potential resource conflicts with the other robots plans. But R_i can try to do more and to adapt its plan to the surrounding activities.
We call this operation *Plan Merging Operation* (PMO) and the resulting plan a *Cooperative Plan*. Such a *Cooperative Plan* (CP_i) consists of a sequence of actions and *execution events* to be signaled to other robots as well as *execution events* that are planned to be signaled by other robots. Such *execution events* correspond to temporal constraints between actions involved in the different coordinated plans.
The PMO (Figure 3 state 2) is performed under mutual exclusion (Figure 3 state 1). R_i collects the plans CP_j^k of the robots which may interfere with IP_i^{k+1}, and builds their union $GP_i^k = \bigcup_j CP_j^k$. Then it performs the merge of IP_i^{k+1} into $GP_i^k : CP_i^{k+1} = PMO(GP_i^k, IP_i^{k+1})$
Various operations on plans can be performed in order to "merge" IP_i^{k+1}. The "only" constraint is that the obtained CP_i^{k+1} is feasible in the current context, and does not introduce any cycle in the resulting GP_i^{k+1}.

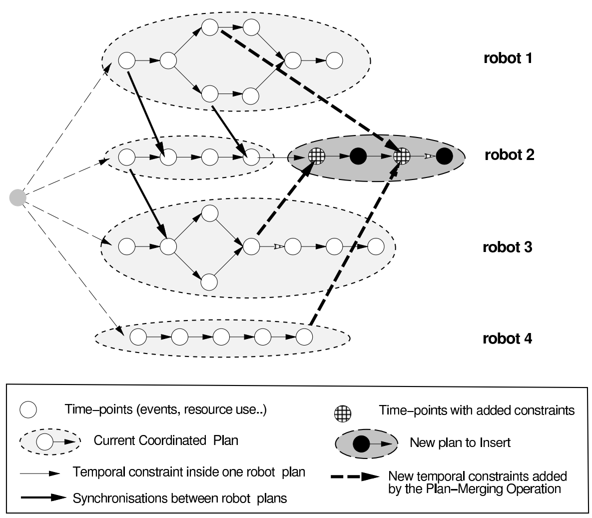

FIG. 4 – Robot 2 performs a PMO.

However a PMO performed by R_i may fail because the final state of at least another robot R_j (as specified in GP_i^k forbids R_i to merge its own plan IP_i^{k+1} in GP_i^k.
There are various ways to deal with such a situation. For instance, the robot may, heuristically, abandon its current goal and the associated plan. While such a reaction may help to avoid the problem, there is no guarantee of convergence nor of global coherence. The next section presents a distributed detection and treatment of such cases that induce, depending on the intricacy of the situation, a progressive transition to more global schemes which

may even ‘degrade” to a unique and centralized planning activity[Qutub et al., 1997].

4.2 Global coherence and deadlock management

Let us call $Pred_i = \{..R_j..\}$ the set of robots that forbids R_i to merge its own plan. In this case, R_i defers its PMO and waits (Figure 3 state 3) until at least one of the robots in $Pred_i$ has performed a new successful PMO which may possibly change the world attributes preventing to merge IP_i^{k+1}. Thus, we introduce temporal order relations between the different plan-merging activities.

Indeed, in addition to *execution events*, i.e. events elaborated by the PMOs and which allow the robots to synchronize their plans, we define *planning events*, i.e. events which occur whenever a robot performs a new successful PMO. The temporal relations between robots plan-merging activities are maintained by each robot R_i in a data structure called *Planning Dependency Graph* PDG_i.

The *Planning Dependency Graph* serves to manage PMOs order (when necessary) as well as to detect *waiting cycles* corresponding to “Merging Deadlock Situations”. The detection of deadlocks during the coordinated decision phase allows execution deadlocks to be anticipated and avoided. Indeed, physical “backtracks” are not always possible or induce inefficient maneuvers.

Dependency Graph Construction. This section focuses on the incremental distributed construction of the Planning Dependency Graph PDG_i and its constraints propagation mechanism.

When R_i starts a new PMO, $Pred_i$ is set to the empty list. If the merging of IP_i^{k+1} in GP_i^k, R_i signals a *planning event* to all robots in $Succ_i$ [4] and clears its current graph PDG_i.

If the PMO has failed, R_i determines $Pred_i$ and checks if it induces planning dependencies which produce cycles in PDG_i(figure 5) :

- If the newly established *planning dependencies* do not introduce any cycle in PDG_i, R_i transmits PDG_i to $Pred_i$.
- If a cycle is created a *planning deadlock situation* is detected which means that the given goals are interdependent and cannot be treated simply by the plan-merging algorithm used, but need to be handled in a single planning step.

When the robot R_k receives PDG_i from R_i, R_k adds it to its own Dependency Graph PDG_k and propagates this new information to all robots in $Pred_k$. R_k is sure that the received PDG_i can be merged with PDG_k without creating any cycle[5].

Deadlock Resolution Strategy. The deadlock resolution strategy that we present is based on a cooperative scheme. We assume that all robots are equipped with a multi-robot

[4]We call $Succ_i$ the set of robots that are directly blocked by R_i.

[5]If such cycle existed, R_i would have discovered it.

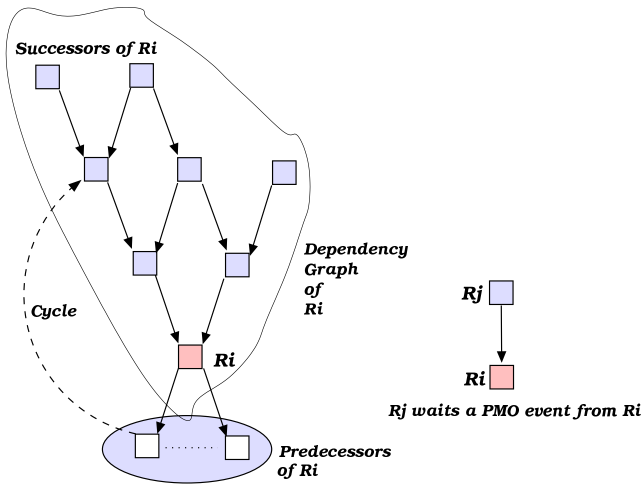

FIG. 5 – Management of Plan Dependency Graphs

planner[6] which can be used, when necessary, for an arbitrary number of robots.

Let us call DL_i^k the set of robots involved in a cycle detected by R_i. When detecting a cycle, R_i has the necessary information in PDG_i to elaborate and validate a plan for all blocked robots in DL_i^k. Note that the blocked robots are unable to add any new executable action to their current coordinated plans CP_j. Therefore, if nothing is done, they will come to a complete stop when their plans CP_j will be completely executed.

To solve the deadlock, R_i becomes the local coordinator (noted R_i^{LC}) for all robots in DL_i^k. To do so, it makes use of its *Local Multi-robot Planner* that will take explicitly, in one planning operation, the conjunction of goals of all the blocked robots (figure 6). This fact will be represented in the Dependency Graph PDG_i as a *Meta-Node* that includes all robots in DL_i^k.

The local coordinator R_i^{LC} must find an *Incremental Multi-robot Plan* (noted IMP_i^{k+1}), if it exists, that solves the conjunction of goals. Once the solution found, R_i^{LC} tries to merge IMP_i^{k+1} into the set of current coordinated plans CP_j of the robots which are not involved in DL_i^k :

$CMP_i^{k+1} = PMO(GP_i^k, IMP_i^{k+1})$ where $GP_i^k = \bigcup_{j \notin DL_i^k} CP_j$

Note that CMP_i^{k+1}, like $IMPi^{k+1}$ is a multi-robot cooperative plan that involve action of all robots belonging to DL_i^k.

- If the merge of $IMPi^{k+1}$ succeeds, R_i^{LC} sends to each robot in DL_i^k its corresponding sub-plan. The meta-node is destroyed and each robot in DL_i^k recovers its initial

[6]Note that it is not strictly necessary to have a multi-robot planner on each robot. A unique multi-robot planner, installed somewhere on the network (at the central station for instance), is sufficient to ensure a correct behavior of the system. The main point, here, is that our scheme is able to determine, in a conservative and incremental way, the set of robots involved in a deadlock and to invoke the multi-robot planner on the set of concerned robots without systematically taking into account all the robots.

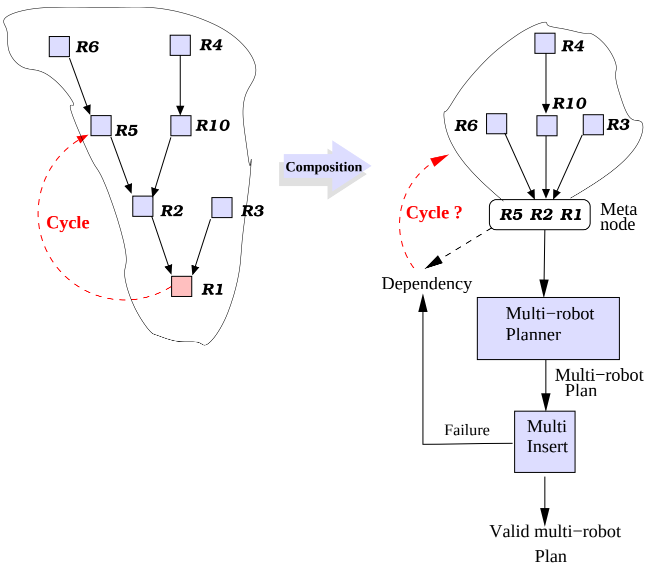

FIG. 6 – Creation of a meta-node in a Plan Dependency Graph

planning and plan-merging autonomy.

– If the merge fails, this means that the final state of at least one robot (not included in DL_i^k) forbids R_i^{LC} to merge $IMPi^{k+1}$. R_i^{LC} determines $Pred_i^{LC}$ and verifies that these newly established constraints do not introduce any cycle in PDG_i^{LC}. In such case, R_i^{LC} defers its PMO, transmits PDG_i^{LC} to all robots in $Pred_i^{LC}$ and waits until one of them has performed a new PMO.
  If a new cycle DL_i^{k+1} is detected, R_i^{LC} generates a new *Meta-Node* containing the union of DL_i^k and DL_i^{k+1} and recursively restarts the same process, acting as a co-ordinator of a greater set of robots.

Note that several deadlocks, which do not interfere, may appear in "parallel" and be solved independently. At the same time, we may have some intricate situations where the *Meta-Node* grows up until it includes the whole system transforming momentarily our distributed system to a completely centralized one (Figure 7).

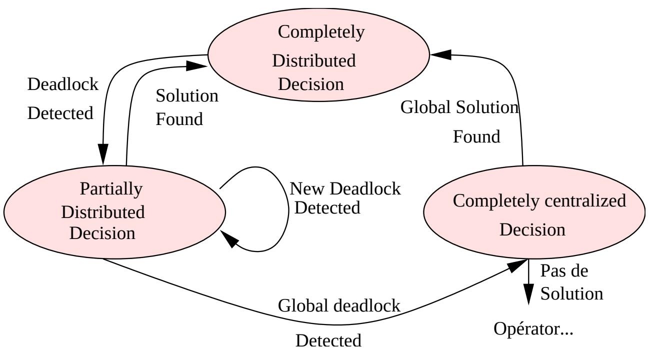

FIG. 7 – Progressive transition to a more global scheme

4.3 Accounting for execution failures

The Plan-Merging paradigm is also robust to execution failures. Indeed, as execution is synchronized through event produced by the robots, when a robot fails in the execution of one of its actions, it is able to inform robots which ask for the occurrence of events it is supposed to produce, that such events will never occur.

This information may cause other robot plans to fail. All robots which have a "broken" coordination plan will rebuild their state and try a PMO again.

Depending on the constraints imposed by an event which will not occur, a cascade of plan failures may occur. This may cause a brutal increase of PMO activities with several robots trying to perform a PMO at almost the same time, but the system will be maintained safe thanks to the properties discussed earlier (guarantee of always having a valid global plan and of detecting deadlocks or situations where a PMO should be deferred).

4.4 General considerations

The are a number of issues that can be discussed within the plan-merging scheme, such as :

– the representation of plans and robot actions
– the operations that can be performed on the plans
– the representation of time, priorities as well as external constraints

For instance, robots may be authorized (or not) to modify the plans that they collect. Indeed, a robot must comply with the "rigid" (non-modifiable) part of the other robot plans, but might be allowed to act on the "flexible" part of the other robot plans.

There are interesting issues such as the definition of new desired features for planners. For example, a useful planner feature can be to synthesize plans that are "easily mergeable".

As already mentioned, we have implemented several instances of the plan-merging paradigm that explore some of these issues. The obtained systems were run on realistic simulation platforms and on real mobile robots (coordinated navigation [Alami et al., 1997]). The overall system allowed a set of autonomous robots not only to perform their tasks in a coherent and non-conflict manner but also to cooperatively enhance their task achievement performance taking into account the robots capabilities as well as their execution context [Botelho and Alami, 2000, Alami and Botelho, 2002].

5 Conclusion

We have discussed a generic architecture for multi-robot operation that provides a framework for cooperative decisional processes at different levels.

Then, we have discussed a coordinated decision scheme called "Plan-Merging Paradigm" that can be used within such an architecture. This paradigm has been designed to control incremental plan adaptation to a multi-robot context.

Various algorithms may be devised to be used within a plan-merging operation, ranging from resource conflict synchronization to more elaborate operations on plans.

Acknowledgments : This work is the fruit of a various contributions from several researchers and particularly : F. Robert, S. Qutub, S. Botelho, F. Gravot, S. Lacroix, J. Gancet, T. Lemaire, G. Hattenberger

I would like also to acknowledge the effective involvement of L. Aguilar, S. Fleury, P. Gaborit, M. Ghallab, M. Herrb, T. Siméon, S. Suzuki.

Références

[Alami et al., 1995] Alami, R., Robert, F., Ingrand, F., and Suzuki, S. (1995). Multi-robot cooperation through incremental plan-merging. In *IEEE ICRA'95*.

[Alami et al., 1997] Alami, R., Fleury, S., Herrb, M., Ingrand, F., and Robert, F. (1997). Multi-robot cooperation in the martha project. *IEEE Robotics and Automation Magazine, Special Issues : Robotics and Automation in Europe*.

[Alami et al., 1998a] Alami, R., Chatila, R., Fleury, S., Ghallab, M., and Ingrand, F. (1998a). An Architecture for Autonomy. *International Journal of Robotics Research*, 17(4) :315–337.

[Alami et al., 1998b] Alami, R., Ingrand, F., and Qutub, S. (1998b). A scheme for coordinating multi-robot planning activities and plans execution. In *ECAI'98*.

[Alami and Botelho, 2002] Alami, R. and Botelho, S. (2002). Plan-based multi-robot cooperation. In Beetz, M., Hertzberg, J., Ghallab, M., and Pollack, M., editors, *Plan-Based Control of Robotic Agents*, volume 2466 of *Lecture Notes in Computer Science*. Springer Verlag.

[Asama and Ozaki, 1991] Asama, H. and Ozaki, K. (1991). Negotiation between multiple mobile robots and an environment manager. In *IEEE ICRA'91*, pages 533–5382.

[Azarm and Schmidt, 1997] Azarm, K. and Schmidt, G. (1997). A decentralized approach for the conflict-free motion of multiple mobile robots. *Advanced Robotics*, 11(4) :323–340.

[Botelho and Alami, 1999] Botelho, S. S. C. and Alami, R. (1999). M+ : a scheme for multi-robot cooperation through negotiated task allocation and achievement. In *IEEE ICRA'99*.

[Botelho and Alami, 2000] Botelho, S. S. C. and Alami, R. (2000). Robots that cooperatively enhance their plans. In *Distributed Autonomous Robotic Systems 4, Lynne E. Parker, George Bekey, and Jacob Barhen (eds.), Springer*.

[Boutilier and Brafman, 1997] Boutilier, C. and Brafman, R. (1997). Planning with concurrent interaction actions. In *AAAI'97*.

[Brumitt, 1996] Brumitt, B. Stentz, A. (1996). Dynamic mission planning for multiple mobile robots. In *IEEE ICRA'96*.

[Burgard et al., 2002] Burgard, W., Moors, M., and Schneider, F. (2002). Collaborative exploration of unknown environments with teams of mobile robots. In Beetz, M., Hertzberg, J., Ghallab, M., and Pollack, M., editors, *Plan-Based Control of Robotic Agents*, volume 2466 of *Lecture Notes in Computer Science*. Springer Verlag.

[Cao et al., 1997] Cao, Y., Fukuna, A., and Kahng, A. (1997). Cooperative mobile robotics : Antecedents and directions. *Autonomous Robots*, 4 :7–27.

[Clement and Durfee, 1999] Clement, B. and Durfee, E. (1999). Top-down search for coordinating the hierarchical plans of multiple agents. In *Third International Conference on Autonomous Agents*, pages 252–259. Association of Computing Machinery.

[Comets-url] Comets project official web page. [Online]. Available : `www.comets-uavs.org`

[Decker and Lesser, 1992] Decker, K. and Lesser, V. (1992). Generalizing the partial global planning algorithm. In *International Journal of Cooperative Information Systems 92*.

[DesJardins et al., 1999] DesJardins, M., Durfee, E., C., O., and Wolverton, M. (1999). A survey of research in distributed, continual planning. *AI Magazine*, pages 13–22.

[Dias et al ;, 2005] B. Dias, R.M. Zlot, N. Kalra, and A. Stentz *Market-Based Multirobot Coordination : A Survey and Analysis* Tech. report CMU-RI-TR-05-13, Robotics Institute, Carnegie Mellon University, April, 2005.

[Dudek, 1997] Dudek, G. (1997). A taxonomy for multi-agent robotics. *Autonomous Robots*, 3 :375–397.

[Durfee and Lesser, 1987] Durfee, E. and Lesser, V. (1987). Using partial global plans to coordinate distributed problem solvers. In *IJCAI87*.

[Ephrati et al., 1994] Ephrati, E., Perry, M., and Rosenschein, J. S. (1994). Plan execution motivation in multi-agent systems. In *AIPS*.

[Gancet 05] J. Gancet, G. Hattenberger, R. Alami & S. Lacroix. *Task planning and control for a multi-UAV system : architecture and algorithms*. IEEE International Conference on Intelligent Robots and Systems (IROS'05), 2005.

[Gerkey and Mataric 04] Brian Gerkey and Maja J Mataric´ *A Formal Framework for the Study of Task Allocation in Multi-Robot Systems* International Journal of Robotics Research, 23(9), Sep 2004, 939-954.

[Gravot and Alami, 2001] Gravot, F. and Alami, R. (2001). An extension of the plan-merging paradigm for multi-robot coordination. In *IEEE International*

Conference on Robotics and Automation, Seoul, Korea.

[Gravot and Alami, 2002] Gravot, F. and Alami, R. (2002). Preliminary results on planning multi-robot cooperative manipulation tasks. In *IEEE IROS 2002, Lausanne, EPFL, Switzerland*.

[Gravot et al., 2003] F. Gravot, S. Cambon, and R. Alami, 'aSyMov : a planner that deals with intricate symbolic and geometric problems', *ISRR*, (2003).

[Jennings, 1995] Jennings, N. (1995). Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence*, 75.

[Laborie, 1995] Laborie, P. (1995). *IxTeT : une approche integrée pour la Gestion de Ressources et la Synthèse de Plans*. PhD thesis, Ecole Nationale Supérieure des Télécommunications.

[Lemai 04] S. Lemai & F. Ingrand. *Interleaving Temporal Planning and Execution in Robotics Domains*. In In proceedings of the National Conference on Artificial Intelligence (AAAI'04), 2004.

[Lemaire 04] T. Lemaire, R. Alami & S. Lacroix. *A Distributed Tasks Allocation Scheme in Multi-UAV Context*. In Proc. of the International Conference on Robotic and Automation (ICRA'04), 2004.

[Mackenzie and Arkin, 1997] Mackenzie, D. and Arkin, R. (1997). Multiagent mission and execution. *Autonomous Robots*, 4 :29–52.

[Mataric, 1994] Mataric, M. (1994). *Interaction and Intelligent Behavior*. PhD thesis, Massachusetts Institute of Technology.

[Parker, 1998] Parker, L. (1998). Alliance : An architecture for fault tolerant multirobot cooperation. *IEEE Trans. on Robotics and Automation*, 14(2) :220–239.

[Parker, 2000] Parker, L. (2000). Current state of the art in distributed robot systems. In *Distributed Autonomous Robotic Systems 4, Lynne E. Parker, George Bekey, and Jacob Barhen (eds.), Springer*, pages 3–12.

[Pollack, 1996] Pollack, M. (1996). Planning in dynamic enviroments : the dipart system. *Advanced Planning Technology : Technology Achievements of the ARPA/Rome Laboratory Planning Initiative*.

[Qutub et al., 1997] Qutub, S., Alami, R., and Ingrand, F. (1997). How to solve deadlock situations within the plan-merging paradigm for multi-robot cooperation. In *IEEE IROS'97*.

[Rosenschein and Zlotkin, 1994] Rosenschein, J. S. and Zlotkin, G. (1994). Rules of and encounter : Designing convention for automated negotiation among computers. *Artificial Intelligence - MIT press*.

[Shoham and Tennenholtz, 1995] Shoham, Y. and Tennenholtz, M. (1995). On social laws for artificial agent societies : off-line design. *Artificial Intelligence*, 0(75) :231–252.

[Simmons 02] R. Simmons, T. Smith, M. Dias, D. Goldberg, D. Hershberger, A. Stentz, and R. Zlot, "A layered architecture for coordination of mobile robots," in *Multi-Robot Systems : From Swarms to Intelligent Automata, Proc. of the 2002 NRL Workshop on Multi-Robot Systems*. Kluwer Academic, 2002.

[Smith, 1980] Smith, R. (1980). The contract net protocol : High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, c-29(12).

[Sullivan et al., 1999] Sullivan, G., Glass, A., Grosz, B., and Kraus, S. (1999). Intention reconciliation in the context of teamwork : an initial empirical investigation. *Cooperative Information Agents III, Lecture Notes in Artificial Intelligence*, 1652 :138–151.

[Tambe, 1998] Tambe, M. (1998). Agent architectures for flexible, practical teamwork. In *First International Conference on Autonomous Agents*.

[Wang and Kumar, 2002] Wang, Z. and Kumar, V. (2002). Object closure and manipulation by multiple cooper ating mobile robots. In *Distributed Autonomous Robotic Systems 5, H. Asama, T. Arai, T. Fukuda and T. Hasegawa (eds.), Springer*, pages 165–174.

[Yuta and Premvuti, 1992] Yuta, S. and Premvuti, S. (1992). Coordination autonomous and centralized decision making to achieve cooperative behaviors between multiple mobile robots. In *IEEE IROS'92*.