A Reconfiguration Algorithm for Distributed Problem Solving

Barbara Dunin-Kęplicz
Institute of Informatics
Warsaw University
Banacha 2, 02-097 Warsaw
Poland
E-mail: keplicz@mimuw.edu.pl
Tel: +48 22 6584594, Fax: +48 22 6583164

Rineke Verbrugge
Cognitive Science and Engineering
University of Groningen
Grote Kruisstraat 2/1, 9712 TS Groningen
The Netherlands
E-mail: rineke@tcw2.ppsw.rug.nl

Abstract

One of the main aspects in Distributed Problem Solving (DPS) in Multi-Agent Systems is that, due to the dynamic and possibly unpredictable environment, team members may fail their tasks or be presented with new opportunities. Thus, it is necessary that team members monitor their performance and replan based on the present situation. This leads to the *reconfiguration problem*.

The paper presents essential aspects of a methodology of teamwork based on the proper maintenance of the individual, social and collective motivational attitudes in a group of heterogeneous agents. The key concept is that of *collective commitment*, formally introduced in [?]. The main phases of DPS, namely construction, maintenance, and realization of collective commitments, are assigned to a four-level abstract architecture. In terms of these generic levels and their interplay an efficient and flexible *reconfiguration algorithm* is specified. Finally, an application of the reconfiguration algorithm in a distributive environment is presented.

1 Introduction

Multi-Agent Systems (MAS) are computational systems in which a collection of (loosely-coupled) autonomous agents interact in order to solve a given problem. As this problem is usually beyond the agents' individual capabilities, agents can communicate, cooperate, coordinate, and negotiate with one another, both in advancement of their own goals as well as for the good of the system as a whole.

Some MAS are referred to as *intentional systems* realizing the practical reasoning paradigm ([?]). The best known and most influential are so-called *belief-desire-intention systems*. BDI-agents are characterised by a "mental state" described in terms of *beliefs*, corresponding to the information the agent has about the environment; *desires*, representing options available to the agent; and *intentions* representing the chosen options.

While beliefs are viewed as agent's *informational* attitudes and desires as its *goals*, intentions and, next, commitments refer to its *motivational* attitudes. From the perspective of teamwork, motivational attitudes are considered on three levels: individual, social (i.e. bilateral), and collective. The process of building up social and collective motivational attitudes starting from individual ones, is formally described in [?]. In the present paper some definitions will be recalled when necessary.

In some recent BDI systems teamwork is modelled explicitly. A dynamic environment poses the problem that team members may fail to bring their tasks to a good end or new opportunities may appear. The explicit model helps the team to monitor its performance and especially to replan based on the present situation. This leads to the so-called *reconfiguration problem* which will be the main subject of this paper.

In order to study the question, we isolate basic stages involved in teamwork in a distributed environment . These are construction, maintenance, and realization of collective commitments. We base our analysis on the four-stage model of [?], containing the consecutive stages of *potential recognition*, *team formation*, *plan formation* and *team action*. These stages will be viewed as levels of abstraction and constitute together an abstract architecture. This architecture will be profitable in clarifying the nature of dependencies between agents involved: some of them will do domain problem solving, others will be responsible for the proper organization of DPS. Taking into account the unpredictable environment, all four stages have a dynamic character and require methods reflecting this. When defining the levels we abstract from particular methods and algorithms meant to realize level-associated goals, but instead formulate their final results and associate them with appropriate individual, social, and collective motivational attitudes.

Especially with respect to collective intentions and collective commitments, our analysis differs from the one in [?]. In fact, in [?] we studied teamwork starting from group formation based on a collective intention, and leading up to the strongest motivational attitude, namely collective commitment. Our approach to collective commitment is plan-based: the ongoing collective intention is split up into subtasks, according to a given social plan. Next, the task allocation is reflected in social commitments between pairs of agents.

Even though the three stages of potential recognition, team formation and plan formation have been extensively discussed in the MAS and AI literature, the important phase of collective team action has received relatively little attention. The requirements of a constantly changing environment lead to the *reconfiguration problem*: when maintaining a collective intention during plan execution, it is crucial that agents replan properly and efficiently when some members do not fulfill their delegated subtasks or are presented with new opportunities. This reconfiguration problem has only recently come to be discussed (see [?], [?]). The main contribution of this paper is the *reconfiguration algorithm* formulated in terms of the abstract levels and their (complex) interplay. An example application of the algorithm is extensively discussed. This research is a continuation of the work presented in [?, ?].

The paper is structured in the following manner. In Section 2 we recall the semantics of individual and social motivational attitudes and the formal definition of a collective commitment. Section 3 presents the four level description of the

BEL(a, φ)	agent a beliefs that φ
GOAL(a, φ)	agent a has as a goal that φ be true
GOAL(a, α)	agent a has as a goal to do α
INT(a, φ)	agent a has the intention to make φ true
COMM(a, b, φ)	agent a socially commits to agent b to make φ true
COMM(a, b, α)	agent a socially commits to agent b to do α
C-BEL$_G(\varphi)$	group G has the collective (mutual) belief that φ
C-INT$_G(\varphi)$	group G has the collective intention to make φ true
C-COMM$_{G,P}(\varphi)$	group G has a collective commitment to make φ true by plan P

Table 1: Individual, social and collective attitudes introduced in [**?**]

abstract system. In terms of these generic abstract levels a general-purpose reconfiguration algorithm is defined in Section 4. Next, an example refinement of this algorithm is discussed in Section 5. Finally, Section 6 presents conclusions and future perspectives.

2 Preliminaries

In Rao and Georgeff's [**?**], individual beliefs, goals, and intentions are formalized as primitive notions and given a formal semantics. We take their semantics as a basis for our formalization of collective motivational attitudes, and refer the reader to [**?**] for details. Extensive descriptions of our own definitions of social commitments and collective intentions, as well as of the well-known notion of collective belief, can be found in [**?**]. In the present paper, short descriptions of social commitments, commitment strategies and collective intentions are repeated when needed. For ease of notation, we use formulas of a multi-modal logic, as in [**?**]. Relevant formulas and their meanings are recalled in table 1.

As a reminder, the temporal structure is a discrete tree branching towards the future, as in [**?**]. The temporal operators include $\mathit{inevitable}(\varphi)$ (in all paths through the point of reference φ holds), $\mathit{optional}(\varphi) \equiv \neg\mathit{inevitable}(\neg\varphi)$, $\Diamond\varphi$ (somewhere later on the same path, φ holds) and $\varphi\ \mathbf{U}\ \psi$ (φ *until* ψ, i.e. either φ holds forever on this path, or, as soon as it stops holding, ψ will hold). Formulas are divided into state formulas (holding in a point) and path formulas (holding on the full path). In [**?**] a similar multi-modal language and the same time structure are used, so we may easily adapt some of their definitions when needed.

Social commitments between two agents were defined in [?], inspired by the discussion in [?]. A social commitment from one agent to the other is not as strong as a collective commitment among them, but stronger than an individual intention, because the other agent is interested in the first one acting on his agreement. Moreover, the agents are aware about the situation, i.e. about their individual attitudes. Such awareness, expressed in terms of collective belief, is generally achieved by communication:

$$\mathrm{COMM}(a, b, \varphi) \leftrightarrow \mathrm{INT}(a, \varphi) \wedge \mathrm{GOAL}(b, \mathit{stit}(a, \varphi)) \wedge \text{C-BEL}_G(\mathrm{INT}(a, \varphi) \wedge \mathrm{GOAL}(b, \mathit{stit}(a, \varphi))),$$

where $\mathit{stit}(a, \varphi)$ means that agent a sees to it (takes care) that φ becomes true.

In[?] our main contribition was a formal definition of collective commitment. As it is the central concept also in this paper, we repeat its characterization here. Inspired by [?] and [?], we treat collective commitment as the strongest motivational attitude in teamwork. A collective intention may be viewed as an inspiration for team activity, whereas the collective commitment reflects the concrete manner of achieving the intended goal by the group. This concrete manner is provided by planning.

We see planning, including means-ends analysis, as a two-step process. The first step is *task division* or decomposition, in which the question is adressed how to decompose a complex task φ into (possibly also complex) subtasks $\varphi_1, \ldots, \varphi_n$. This step is followed by *task allocation*, in which actions are associated to the subtasks constructed in task division, and these are given to team members. This results in pairs $< \alpha_i, a >$ of a (possibly complex) action α_i that realizes task φ_i and an agent a. To make a social plan complete, the temporal structure among these pairs should be established.

A plan may be constructed immediately as a completely instantiated one, where to every individual member of the team some specific tasks are allocated in a completely specified temporal structure. On the other hand, recursive planning is also possible, where tasks may be decomposed and allocated recursively. At the end of the recursive process, the result is a completely instantiated plan. Both kinds of plan can be denoted by a *social plan expression*. Here follows a recursive definition of well-formed social plan expressions, adopted from [?] and inspired by dynamic logic.

- If p is an atomic action and y is an agent, then $< p, y >$ is a well-formed social plan expression;
- If φ is a well-formed state formula and y is an agent, then $<!\varphi, y>$ (standing for y to *achieve* φ) and $<?\varphi, y>$ (standing for y to *test* the truth of condition φ) are well-formed social plan expressions;
- If α and β are well-formed social plan expressions, then $< \alpha;\beta >$ (sequential composition), $< \alpha \parallel \beta >$ (parallellism) and $< \alpha \mid \beta >$ (non-deterministic choice) are well-formed social plan expressions.

The last part of the above definition introduces the temporal relations between the execution of subtasks. The plans on which collective commitments are based are always represented as social plan expressions.

A collective commitment among a group G based on a social plan P can only be established or maintained if the group has the associated collective intention. In addition, for every one of the subgoals $\varphi_1, \ldots, \varphi_n$ that together constitute the goal φ in question reflected in the plan P, there should be one agent in the group who is socially committed to at least one (mostly other) agent in the group to fulfil the subgoal. Moreover, there should be a common belief in the whole group that the plan will be entirely realised, i.e. that all subgoals have been adopted by committed members of the group. The defining axiom below reflects all these characteristics.

$$\text{C-COMM}_{G,P}(\varphi) \leftrightarrow \text{C-INT}_G(\varphi) \wedge \bigwedge_{\varphi_i \in P} \bigvee_{a,b \in G} \text{COMM}(a,b,\varphi_i) \wedge$$
$$\text{C-BEL}_G(\bigwedge_{\varphi_i \in P} \bigvee_{a,b \in G} \text{COMM}(a,b,\varphi_i))$$

Let us stress that in this definition the team as a whole is aware that social commitments between team members have been established in the proper way, without explicit knowledge of each bilateral commitment. This kind of explicit knowledge can be incorporated when needed.

When defining the notion of collective commitment we intended to take only necessary ingredients into account. However, if any extension is needed, other factors may be easily incorporated into our framework by adding extra axioms. The semantics of the system is defined similarly as in [?], except that we have different definitions for the collective attitudes.

3 The four level abstract description of the system

The analysis of DPS within MAS led us to the distinction of four levels of abstraction, corresponding to the four stages defined in [?]. When characterizing them, we will take the perspective of a BDI architecture: the key point is to bind particular levels of abstraction with appropriate social, collective and individual motivational attitudes. We suppose that the main goal of the system is given at the beginning. The problem of goal selection is beyond the scope of this paper and has been extensively treated elsewhere [?].

3.1 The four levels

- Level 1: the *potential recognition* level.
  Analogous to [?], we consider DPS to begin when some agent in a multi-agent environment recognizes the potential for cooperative action in order to reach his goal. The input of this stage is an agent a, a goal φ plus a finite set T of agents from whom a potential team may be formed. The output at this stage is the "potential for cooperation" that agent a sees with respect to φ, to be defined below by an analog of the definition of [?]. A few preliminary definitions are needed, which can be looked up in table 2. Most of these are also taken over from [?] and [?], except that we prefer non-recursive definitions of individual and group ability (CAN and C-CAN_G). Our choice is motivated by the need to make revision explicit in the algorithm presented in the next section. Moreover, we introduced a new predicate $\text{STRAT}(\varphi, G)$ reflecting possible combinations of commitment strategies suitable for achieving φ. The definitions of agents' commitment strategies, addressing the question whether and in which circumstances the agent is allowed to responsibly drop a social commitment, are introduced in [?]. Inspired by those in [?], they additionally obey the social aspects of communication and coordination. Below, the definitions are repeated.

  The strongest commitment strategy is followed by the *blindly committed* agent, who maintains its commitments until it actually believes that they have been achieved. Single-minded agents may drop social commitments when they do not believe anymore that the commitment is realizable. However, as soon

$\mathrm{KNOW}(a, \varphi)$	agent a knows that φ
$\text{C-KNOW}_G(\varphi)$	group G has the collective (common) knowledge that φ
$\mathrm{AGENTS}(\alpha, G)$	group G is precisely the set of agents required to perform the action α;
$\mathrm{STRAT}(\varphi, G)$	group G have a suitable distribution of commitment strategies to achieve φ;
$happens(\alpha)$	action α happens next on the current path;
$\mathrm{ACHIEVES}(\alpha, \varphi)$	$inevitable(happens(\alpha) \to happens(\alpha; \varphi?))$ i.e. on all paths, after action α happens, proposition φ is true;
$\mathrm{CAN}(a, \varphi)$	$\exists\alpha(\mathrm{KNOW}(a, \mathrm{AGENTS}(\alpha, \{a\}) \wedge \mathrm{ACHIEVES}(\alpha, \varphi)))$ i.e. there is a complex action α such that a knows that a itself is the single agent required to perform α, and that α achieves φ;
$\text{C-CAN}_G(\varphi)$	$\exists\alpha(\text{C-KNOW}_G(\mathrm{AGENTS}(\alpha, G) \wedge \mathrm{ACHIEVES}(\alpha, \varphi)))$.

Table 2: Formulas needed for the definition of potential recognition

as the agent abandons a commitment, some communication and coordination with the other agent is needed. For open-minded agents, the situation is similar as for single-minded ones, except that they can also drop social commitments if they do not aim for the respective goal anymore. As in the case of single-minded agents, communication and coordination will be involved.

As was shown in [**?**], the agents' commitment characteristics, reflected in their commitment strategies, predispose them to do different tasks. We suggest that agent a take these characteristics into account, starting already from potential recognition, reflected in the conjunct $\mathrm{STRAT}(\varphi, G)$ below.

Now we are ready to give the definition for "agent a sees the potential for cooperation to achieve goal φ":

$$\begin{aligned}&\mathrm{POTCOOP}(a, \varphi) \leftrightarrow \mathrm{GOAL}(a, \varphi) \wedge \\ &\quad \exists G \subseteq T\mathrm{BEL}(a, \text{C-CAN}_G(\varphi) \wedge \mathrm{STRAT}(\varphi, G)) \wedge \{\neg\mathrm{CAN}(\alpha, \varphi) \vee \\ &\quad \mathrm{BEL}(a, \neg\exists\alpha[\mathrm{AGENTS}(\alpha, \{a\}) \wedge \mathrm{ACHIEVES}(\alpha, \varphi) \wedge \mathrm{GOAL}(a, \alpha)])\}\end{aligned}$$

More informally: φ is a goal of a, and there is a group G such that a believes that G can jointly achieve φ and that they have the right distribution of commitment strategies to do so; and either a cannot or doesn't desire to achieve φ in isolation. As output of a successful outcome of this stage, track is also kept of the sequence $(G_1, \ldots, G_n)$ of all relevant groups $G \subseteq T$ for which the condition $\mathrm{BEL}(a, \text{C-CAN}_G(\varphi) \wedge \mathrm{STRAT}(\varphi, G))$ was found to be true in the establishment of $\mathrm{POTCOOP}(a, \varphi)$.

- Level 2: the *team formation* level.
  Suppose that agent a sees the potential for cooperation to achieve φ. Somewhat different from [**?**], we find that during the team formation stage agent a attempts to bring it about in some group G that the group has the *collective intention* to make φ true. The input of this stage is agent a, a formula φ and sequence of potential groups $(G_1, \ldots, G_n)$ as output by stage 1. The output of a successful outcome of this stage is one group G from $(G_1, \ldots, G_n)$ together with a collective intention among G to achieve φ, which includes corresponding individual intentions of all group members.

  Let us recall from [**?**] the definition of a collective intention. A necessary condition is that all members of the group have the associated individual intention, and that it is a common belief in the group that all members have this intention, i.e. all members of the group are aware of this. Moreover, all members in the group should *intend* the other members to have the associated individual intention, and it should be a common belief in the group that this is so.

In order to formalize the above two conditions, $\text{E-INT}_G(\varphi)$ (standing for "everyone intends") is defined by $\text{E-INT}_G(\varphi) \leftrightarrow \bigwedge_{i \in G} \text{INT}(i, \varphi)$. Now we are ready to repeat the axiom defining collective intentions:

$$\begin{aligned} \text{C-INT}_G(\varphi) \leftrightarrow \quad & \text{E-INT}_G(\varphi) \wedge \text{C-BEL}_G(\text{E-INT}_G(\varphi)) \\ & \wedge \text{E-INT}_G(\text{E-INT}_G(\varphi)) \\ & \wedge \text{C-BEL}_G(\text{E-INT}_G(\text{E-INT}_G(\varphi))) \end{aligned}$$

To summarize, at the team formation level a group G and the collective intention of the group, which includes corresponding individual intentions of all group members, are established. This is done by subsequently attempting to establish the collective intention among G_1, G_2 etc., until this succeeds for some G_i. The sequence of still untried potential groups $(G_{i+1}, \ldots, G_n)$ is stored for revision purposes.

- The *plan generation/formation* level.
  This level includes *task division* and *task allocation*. The input of this stage is a team G together with its collective intention to achieve a goal φ. During a successful run of this stage, first, an adequate task division of φ into a sequence of subtasks $\varphi_1, \ldots, \varphi_n$ is constructed, ensuring the certain realization of the collective goal. Subsequently, an appropriate subtask allocation to the members of the team needs to be established and a temporal structure among the tasks devised. By 'appropriate allocation' we mean that not only agents' abilities and resources, but also their commitment strategies are taken into account. The result of this two-step process is a social plan P (see section 2). Note, however, that there is a close interplay between task division and task allocation, and that the above-mentioned characteristics of the team members play an important role already at the substage of task division. We do not elaborate here on the ways by which the final social plan is constructed - this subject has been exhaustively discussed in AI literature (see e.g. [**?**]). For the complex process of negotiation that comes to the fore in plan generation, see [**?**].

  In effect, at the plan generation level all agents from a group socially commit to carry out their respective subtasks, and to communicate about these social commitments in order to establish pairwise mutual beliefs about them. This information exchange concludes the *collective* part of plan generation. The final successful outcome of this stage is collective commitment $\text{C-COMM}_{G,P}(\varphi)$ of the group G based on the social plan P. Because of our strong notion of collective commitment, this definition of the successful outcome of the stage of plan generation is somewhat stronger than the one in [**?**].

- The *plan execution / team action* level.
  This level includes *execution of actions* and the *reconfiguration procedure*. At the level of plan execution, all team members aim at realizing their own subtasks from the collective commitment $\text{C-COMM}_{G,P}(\varphi)$ constructed at stage 3. Thus they start executing adequate agent-specific actions. Many different situations may occur, some of which imply reconfiguration among the group, an aspect which is not treated explicitly in [**?**]. We will discuss reconfiguration in the sequel. In terms of motivational attitudes, plan realization amounts to the maintenance of *social commitments* and associated *individual intentions*.

The successful outcome of the stage of plan execution is that all subtasks making up the social plan P have been carried out by the agents who were socially committed to do them, reflected in the team's collective commitment and that by the success of their actions the goal φ of the collective commitment has been achieved.

Now we are ready to define success and failure of the system as a whole. The *system fails* if the overall goal φ is not realizable by any team from the relevant set T; the *system succeeds* otherwise.

3.2 The structure of the system

The diversity of the system's functions should be reflected in the structure of the system: in addition to agents realizing the domain problem solving there need to be agents aiming at the proper *organization* of DPS, all of them operating on different levels. The goals of such organizing agents could be the following:

- To organize the control on particular levels; let us note that here such complex tasks like cooperation, communication and negotiation are involved.
- To realize the reconfiguration method as the interplay between different levels.

Let us stress that introducing organizing agents does not intend to restrict the MAS paradigm; it is meant as a tool allowing proper realization of ideas presented here.

4 The reconfiguration method

The dynamics of the system will be reflected in the *cycle of the system*. In order to achieve a given overall goal, one cycle realizes the following consecutive steps: potential recognition, team formation, plan generation, and plan execution.

In the perfect case a group achieves the goal in the way it was planned from the very beginning. In the more common non-perfect case disturbances appear in the first cycle of the system at some level. In such a situation some kind of *reconfiguration* is necessary. Because the need to reconfigure may appear at every moment, the cycle of the system will be referred to as the *reconfiguration algorithm*. The main purpose of the algorithm is *the proper maintenance of collective commitments*. To properly deal with the variety of situations the reasons of disturbances have to be recognized. Therefore, a few definitions will be introduced.

An action execution fails for an *objective reason* if the action is not realizable by anybody in the present team in the current state of the world.

An action execution fails for a *subjective reason* if the agent to whom it is delegated does not believe that he can achieve it.

The problem of choosing adequate properties of a reconfiguration method may be viewed as an open question, possibly related to the type of application considered. When formulating the reconfiguration algorithm, we chose some intuitive properties corresponding to classical strategies adopted in backtracking. We postulate that the system behaviour should preserve *continuity*. This means that, if an obstacle appears, the problems are solved by moving up in the hierarchy, but as little as possible. Thus, one moves to the nearest point up in the hierarchy of levels where a different choice is possible. If such a point does not exist anymore, the reconfiguration algorithm fails. In other words, depth-first search is used.

The continuity criterion determines the overall structure of the algorithm.

4.1 Conservativity

The continuity criterion is context-independent. As the basis of local decisions a context-sensitive criterion would be valuable. When new rounds at particular levels are necessary, the question arises which results should be preferred based on the failed ones. The answer to this question is certainly context-dependent and requires formulation of a specific notion of the distance between teams, goals, plans, etc. It seems that for a wide class of application domains, the system should behave in a rather *conservative* way.

In our case, conservativity (or *inertia*) entails that the collective commitment that is being carried out should change as little as possible. In particular, conservativity includes:

- When *team formation* is not realizable for the current sequence of teams, a new round of potential recognition is needed, taking care that the new sequence of potential teams is as close as possible to the previous one.

- If a *task division* is not realizable by the current team a round of team formation is needed in order to create a new team for the current collective intention, taking care that as few as possible team members are replaced by new ones.

- When making a new *task division* for the current collective intention and the team, one should take care that as few as possible tasks are replaced by new ones. In particular, one should try to reinclude the tasks that have been carried out already into the new plan, so that hard work is not thrown away unnecessarily.

- When making a new *task allocation* for the current task division one must take care that as few as possible tasks (and only subjectively failed ones) are reassigned.
- During replanning as a whole, one should take care that the new social plan is as close as possible to the previous one, taking into account tasks that have already been successfully achieved. This entails that even in the case of a new task division, one should remember the previous task allocation and give "old" (or similar) tasks to agents that were assigned them before. Thus, division and allocation should not be viewed independently.

4.2 Reconfiguration algorithm

The (informal) description of the reconfiguration algorithm below is meant to be generic: a pattern of behavior is described in terms of abstract level-associated procedures (i.e. *potential-recognition, team-formation, task-division, task-allocation, plan-execution*), without fixing any particular method or strategy. Input and output parameters, as well some other conditions of these complex procedures are commented in the algorithm itself. Let us stress, that each of these procedures may succeed or fail - in this sense the structure of each abstract level is analogical.

To make the algorithm's structure more transparent, we decided to bind a label with each level-associated procedure and to use the *goto* statement.

In the algorithm an assignment of new social and individual motivational attitudes is made to create a new collective commitment. In this sense this algorithm may be viewed as a *revision* of motivational attitudes. In the body of the algorithm phases of belief revision and motivational attitudes revison are distinguished, without further refinement. They are realized by abstract procedures *belief-revision* and *motivational-attitudes-revision*, after *motivational-attitudes-assignment*. Let us note however, that in the description of the algorithm we have resigned from explicitly splitting collective motivational attitudes into individual and social ones. Also, for the sake of simplicity, the above procedures haven't fixed the number of paramethers, but their proper use is always implied by the context.

From a pragmatic point of view, the proper treatment of revision is possible, because it is assumed that agents are obliged to communicate about changes. On the other hand, the presence of social commitments and common beliefs about them *only* between partners for whom they are relevant, together with the conservativity assumption, ensures that motivational attitudes revision is as efficient as possible.

Let us stress that system failure and success are realized by complex procedures *system-failure* and *system-success*, respectively. It is also assumed that all the required information is available at the development-time.

Reconfiguration algorithm:

```
begin
{input of the system:
Q - a goal of agent S;
S - the agent that is input to the system and that will recognize potential;
T - a set of agents from whom potential teams are selected}
A: potential-recognition (Q, S, H, T);
   {input: Q, S, T}
   if not (potential-recognition-succeeded) then
      {S does not see any potential for cooperation with respect to goal Q}
      system-failure;
      STOP
   else
   {potential recognition succeeded - output: H - a sequence of teams H =(G_1, ..., G_n)
```

```
    with the potential to realize Q}
   initialization-of-motivational-attitudes(Q,S);
   {the attitude POTCOOP(S,Q) is established}
B: team-formation (Q, S, H, G);
   {input: Q, S, H}
   if not (team-formation-succeeded) then
   {the collective intention w.r.t. Q cannot be established among any of the teams from
H;
    return to the potential recognition level for S to construct a new sequence of potential
     teams for which it sees cooperation potential w.r.t. Q}
      goto A
   else
   {team formation succeeded - output G: a set of agents aiming to realize Q; Suppose
G is the
      first unused G_i from H for which the collective intention towards Q can be
established}
   motivational-attitudes-assignment (Q, G);
   {the collective intention C-INT_G(Q) of G towards Q is established here}
C: task-division (Q, G, R);
   {input: Q, G}
   if not (task-division-succeeded) then
   {task division failed; return to the team-formation level in order to attempt the first
unused
     G_{i+1} from H to establish collective intention towards Q}
   goto B
   fi;
   {task division succeeded - output: R - a sequence of subtasks together realizing Q;
    i.e. the first part of a social plan P}
D: task-allocation (Q, G, R, P);
   {input: Q, G, R}
   if not (task-allocation-succeeded) then
   {task allocation failed; return to the task division level}
     goto C
   fi;
   {task allocation succeeded - output: a social plan P corresponding to subtask
sequence R}
      motivational-attitude-assignment (Q, G, R, P);
      {the collective commitment C-COMM_{G,P}(Q) among G to goal Q with respect to
social
        plan P is established (including corresponding social commitments to subtasks
from R)}
      {plan execution part starts here}
   E: plan-execution (Q, G, R, P);
      {input Q, G, R, P}
      if plan-execution-succeeded then
      {all actions that constitute the plan P are successfully executed; agents' beliefs
and
        motivational attitudes need to be revised to reflect that the overall goal Q is
achieved
        as well as agents' subgoals from R}
         belief-revision (Q, G, R);
         motivational-attitudes-revision (Q, G, R, P);
         system-success;
         STOP
      elseif
         {some action execution failed: differentiation of reasons for failure}
         F1:if subjective-reason-for-failure(Q, P, G, R, R1, R2) and
         not (objective-reason-for-failure(Q, P, G, R, R1 ,R2)) then
```

{R1: the sequence of tasks from R that have been successfully achieved thus far;
R2: the sequence of pairs (A, X) of tasks that failed plus reasons for failure, where X=ob or X=sub}
{for every action that failed, it failed for a subjective reason, i.e. $\forall A \in R\, \forall X((A,X) \in R2 \rightarrow X = sub)$}

```
belief-revision (G, R, R1, R2));
if task-reallocation-possible then
```

{for every action that failed for a subjective reason, there is another team member believing it can achieve it, i.e. $\forall A \in R\, \forall M_1, M_2 \in G((A, sub) \in R2$ $\wedge \mathrm{COMM}(M_1, M_2, A) \rightarrow \exists M_3 \in G(M_1 \neq M_3 \wedge \mathrm{BEL}(M_3, \mathrm{CAN}(M_3, A))))$; before an attempt at task reallocation based on P, R1 and R2 is made, a revision of motivational attitudes is needed}

```
        motivational-attitudes-revision (Q, P, G, R);
        goto D
    elseif
```

{no task reallocation is possible: a new task division is needed}

```
        goto C
    fi
fi
```

{there are also objective reasons for failure, i.e. $\exists A \in R((A, ob) \in R2)$}

```
F2:  belief-revision (G, R, R1, R2);
```

{now it should be investigated whether the objective reason for failure blocks achieving the overall goal Q}

```
if blocked(Q) then
```

{the objective reason for failure blocks achieving the goal Q}

```
    system failure;
    STOP
else
```

{the objective reason for failure does not block achieving the goal Q; a new task division is needed, return to the task division level}

```
    goto C
  fi
fi
end
```

Let us trace the cycle of the system during a realization of the reconfiguration algorithm. When consecutive steps of reconfiguration go well, changes of individual, social and collective motivational attitudes are commented in the description of the algorithm. Let us focus on the points where failure of the four main stages of the process takes place.

The failure of the potential recognition level (see the label A) meaning that agent S does not see any potential for cooperation with respect to the goal Q, leads to the total failure of the system.

The failure of the team formation level (see the label B), meaning that the collective intention with respect to Q cannot be established among any of the teams from H, requires a return to the potential recognition stage to construct a new sequence of potential teams.

The failure of the task division level (see the label C) requires a return to the team formation level in order to establish a collective intention in the chosen new team from H and may be viewed as the reconfiguration of the team together with a change of motivational attitudes on the level of collective intention and respective individual attitudes.

The failure of the task allocation level (see the label D) requires a return to the task division level in order to create a new sequence of tasks. This may be viewed as the reconfiguration of the first part of a social plan P to realize the goal Q.

When, finally, a collective commitment is established, the failure of some actions together constituting the social plan P requires the reconfiguration of motivational attitudes on the level of collective commitment.

4.3 Complexity of the reconfiguration algorithm

Let us recall that the reconfiguration algorithm is an abstract one. Thus, it is meant to be instantiated for each particular application.

When defining the levels we abstracted from particular methods and algorithms meant to realize level-associated goals, but instead formulated their final results and associated them with appropriate motivational attitudes. Let us stress that the procedures realizing level-associated goals are rather complex.

Especially the plan generation / formation level, including task division and task allocation, is complex from both the AI and the MAS perspective. As for the first perspective, planning is involved. If done from first principles, planning is in general undecidable [?], while for limited domains it may still be tractable. If, on the other hand, a plan library is used, searching the library may be complex as well. As for the MAS perspective, the aspects of communication, negotiation and coordination come to the fore on the level of plan generation. Even though the work on negotiation is still in progress, it is quite clear that the procedures being proposed are usually rather complex [?]. The same holds for coordination.

On the level of plan execution, belief revision, which is known to be NP-hard, is repeatedly performed. Also, if there are objectively failed tasks, it needs to be checked whether their failure blocks the overall goal. This kind of consistency check is co-NP complete. Let us stress that all the complex aspects mentioned here have to be treated in any rigorous approach to the reconfiguration problem.

As for the global structure of the reconfiguration algorithm, it is based on backtracking search. In the generic case, when not using any domain-dependent information, context-dependent improvements as obtained in informed search methods cannot be made.

However we did use forward checking in the algorithm in the check whether an objectively failed task blocks the overall goal. Next, in order to be certain that the search method is complete and optimal, iterative deepening may be used. It doesn't increase the time or space complexity when compared with depth-first search. Due to its exponential space complexity, breadth-first search is no option for reconfiguration.

We would like to stress that our definition of collective commitment ensures efficiency of reconfiguration in two ways. Firstly, all the motivational attitudes occuring in the definition of collective commitment are defined in a non-recursive way. This allows straightforward revision of all relevant motivational attitudes. Secondly, only pairwise social commitments to subtasks appear in the definition of collective commitment, so that replanning and motivational attitudes revision are made less complex. Thus, in contast to [?], using our definition of collective commitment it is often sufficient to revise some of the pairwise commitments, instead of involving the entire team in the replanning process.

5 Example of distributed problem solving

In this section, an example of distributed problem solving is shown in which the reconfiguration algorithm is implemented profitably. The overall goal of the system is to prove a new mathematical theorem. In this application domain, we decided to use the conservativity assumption.

The description of the example will not be too precise, e.g. we will not give formal rules governing mathematical provability. We will follow the cycle of the system as given in the reconfiguration algorithm.

5.1 Potential recognition

We assume that all agents are prepared to work in a distributed environment, communicating, coordinating and negotiating when necessary. Let us suppose that the system starts with an open-minded agent t who looks for cooperation potential for the overall goal $prove(theorem(T))$, from among the set of agents $\{t, l, l_2, c\}$ where both l and l_2 are single-minded agents of whom t believes that they are able to prove the lemmas needed for theorem T, and c is a single-minded agent of whom t believes that it can check all the proofs that will be constructed. Moreover, t cannot prove T on his own, but there is a sequence of potential teams of which t believes that each team can jointly achieve the goal, namely $H =< \{t, l, c\}, \{t, l_2, c\}, \{t, l, l_2, c\} >$. Thus $\text{POTCOOP}(t, prove(theorem(T))$ is established.

When recognizing the potential for cooperation, we postulate to take, among other aspects, agents' commitment characteristic into account. As discussed in [**?**], blindly committed agents are not at all adaptive. Because in research, plan changes are frequently necessary, it is crucial that the percentage of blindly committed agents in the team is not too high, and that enough open-minded agents are present to pitch in if necessary. Thus, in H above, all three teams contain only one blindly committed member and at least one open-minded one.

5.2 Team formation

During team formation, a team should be formed to adopt the collective intention of proving theorem T. After the potential recognition phase, agent t first tries to get the collective intention established in the first team from his potential team sequence H, namely $G = \{t, l, c\}$. In this case, let's suppose that t succeeds and $\text{C-INT}_G(prove(theorem(T))$ is established. (Otherwise t has to successively try to establish the collective intention in the other two teams.)

5.3 Plan generation

During *task division* a sequence of lemmas is determined from which the theorem T should be proved. More formally, given the theorem T to be proved, there are different divisions $L_0, L_1, \ldots$ of the theorem into sequences of lemmas, where $L_i = (l_1^i, \ldots, l_{d(i)}^i)$.

On the basis of this task division, the next step is *task allocation*: specific subtasks are to be allocated to adequate agents having resources and skills sufficient to realize the allocated goal, as well as suitable commitment strategies. In this case, according to their abilities, the open-minded agent t (theorem prover) is allocated to prove the theorem T from sequences of lemmas, the single-minded (lemma prover) l is allocated to prove lemmas, and the blindly committed c (proof checker) is allocated to check the others' proofs of lemmas and of the theorem. Below follows a sample task division and allocation, corresponding to the first division $L_0 = (l_1^0, \ldots, l_n^0)$ of the theorem into lemmas.

- $prove(theorem(T))$: to prove the theorem (by the whole group);
- $prove(lemma(l_i^0))$: to prove the specific lemmas $l_1^0, \ldots, l_n^0$ (by lemma prover l);
- $check(proof(l_i^0))$: to check all specific proofs of the corresponding lemma l_i^0 (by proof checker c);
- $prove(theorem\text{-}from(L_0))$: to prove the theorem from the sequence of lemmas $L_0 =< l_1^0, \ldots, l_n^0 >$ (by theorem prover t);
- $check(proof\text{-}from(L_0))$: to check all proofs of the theorem from the sequence of lemmas L_0 (by proof checker c).

Next, we are ready to give the succesfull condition for a proof of the full theorem:

> There is a division of the theorem T into lemmas such that for all lemmas in the sequence there exists a proof, constructed by the lemma prover, that has been found correct by the proof checker. Also there is a proof of the theorem from the sequence of lemmas, constructed by the theorem prover, which has been found correct by the proof checker.

This condition forms the basis for a simple sequential social plan to realize the collective intention of proving the theorem:

$$P =<< prove(lemma(l_1^0)), l >; < check(proof(l_1^0)), c >; \ldots;$$
$$< prove(lemma(l_n^0)), l >, < check(proof(l_n^0)), c >;$$
$$< prove(theorem\text{-}from(L_0)), t >; < check(proof\text{-}from(L_0)), c >>.$$

In practice, a different temporal structure may be more efficient, e.g. subtasks could more efficiently be carried out in parallel. Based on this plan, the group is prepared to establish a *collective commitment.* It is made up of the following conjuncts, relative to the plan P and to $G = \{t, l, c\}$:

$$\begin{aligned}
&\text{C-COMM}_{G,P}(prove(theorem(T))) \leftrightarrow \text{C-INT}_G(prove(theorem(T))) \wedge \\
&\quad \bigwedge_{i=1}^{n} (\text{COMM}(l, t, prove(lemma(l_i^0))) \wedge \text{COMM}(c, l, check(proof(l_i^0)))) \wedge \\
&\quad \text{COMM}(t, l, prove(theorem\text{-}from(L_0))) \wedge \text{COMM}(c, t, check(proof\text{-}from(L_0))) \wedge \\
&\quad \text{C-BEL}_G(\bigwedge_{\alpha \in P} \bigvee_{a,b \in G} \text{COMM}(a, b, \alpha))
\end{aligned}$$

In the sense of goals to be reached, no more social commitments are needed, although communication about these goals is always possible, for example scientific discussion. The common beliefs appearing in the definition of collective commitments are established by bilateral communication between the agents involved.

5.4 Plan execution

The actions to be carried out by the agents in the team correspond directly to subtasks introduced in the plan P and reflected in the collective commitment $\text{C-COMM}_{G,P}(prove(theorem(T)))$. During plan execution, the degree of responsibility and thus complexity of the three agents is quite different. The blind agent can work on his delegated tasks without tracking what happens with the other agents and without pitching in for them. However, the single-minded agent and especially the open-minded agent have to react when the agent(s) working for them fail to bring their tasks to a good end.

Below we sketch the reconfiguration procedure for the example, corresponding to section 4. As in the general case, when a particular action fails it is important to recognize that each reason gives rise to a different adequate reaction satisfying continuity, of which the reconfiguration procedure takes care, For this example we also choose to make local decisions obeying conservativity. Usually the agent for whom the action is executed reacts. Suppose that in the current round, the team is proving the theorem T from lemma sequence $L_j =< l_1^j, \ldots, l_{d(j)}^j >$.

The reaction to failure in the reconfiguration algorithm can be subdivided into two different cases, first according to different *subjective reasons* for failure of actions: part F1 of the reconfiguration algorithm. Actions that fail for subjective reasons happen when:

sub 1 The proof checker c doesn't finish $check(proof(l_i^j))$ before a pre-set time limit. In this case, the most efficient and conservative way of task reallocation is to let the single-minded lemma prover pitch in for c by checking his own proof, and keep the rest of the plan unchanged. Thus, first belief revision in group G is needed, then a new individual intention of l to $check(proof(l_i^j))$ is added, as well as a social commitment toward c to do this, which is, finally, reflected in a new collective commitment.

sub 2 The lemma prover l does not believe he can carry out $prove(lemma(l_i^j))$, but without knowing a counter-example. The most conservative way of task reallocation is to let t pitch in for l if t believes that he can prove the lemma. After the belief revision in group G, agent l drops his social commitment towards t and his individual intention to make the proof, which are both taken over by t towards him and reflected in a new collective commitment.

sub 3 The theorem prover t does not believe that he can carry out $prove(theorem\text{-}from(L_j))$, but without knowing a counter-example. The most conservative way of task reallocation is to let l pitch in for t if l believes that he can prove the theorem. After the belief revision in group G, agent t drops his social commitment towards l and his individual intention to make the proof, which are both taken over by l towards him and reflected in a new collective commitment.

In part F2 of the reconfiguration algorithm, objective reasons of failure are considered. First, following the reconfiguration algorithm, it needs to be checked if the overall goal $prove(theorem(T))$ is blocked. In the example, this happens only in case a counterexample to theorem T is found by c, in which case the system fails.

There are also two cases which do not cause blocking of the overall goal; in both cases, following the reconfiguration algorithm, a new task division has to be made. Here follows an elaboration of both cases.

ob 1 The proof checker finds a mistake in a proof; then the action of making the proof has failed, but may be repeated by constructing a new proof of the same lemma or of the theorem. In this case, to obey conservativity, almost the same division of lemmas and the same task division and later allocation are proposed as in the present round. The only change is that the maker of the faulty proof gets as new task to make a new proof, which is then reflected in the social plan and in a social commitment such as $\mathrm{COMM}(l, t, prove\text{-}lemma\text{-}diff(l_i^j, p))$ (l commits to t to make a new proof of lemma l_i^j, different from the faulty proof p), and finally in a new collective commitment.

ob 2 The proof checker finds a counterexample to a lemma; then the action of making the proof has failed, and it has no sense to try again. At the new round of task division, the following happens. In case a counterexample to lemma l_i^j in the sequence of lemmas L_j was found, a new division of the theorem into a lemma sequence L_{j+1} has to be made. To obey conservativity, the new lemma sequence should resemble L_j as closely as possible. Especially important is to conserve as many as possible lemmas from the initial sequence $l_1^j, \ldots, l_{i-1}^j$ which have already been proved and checked. (Thus hard work is not thrown away unnecessarily.) Thus a new sequence of tasks is made corresponding to L_{j+1}.

During task allocation, the task of proving the lemmas is again allocated to l, and the task of proving the theorem from lemma sequence L_{j+1} is allocated to t. Agent c is again allocated to check all proofs. The resulting social plan is:

$Q =<< \mathit{prove}(\mathit{lemma}(l_1^{j+1})), l >; < \mathit{check}(\mathit{proof}(l_1^{j+1})), c >; \ldots;$
$< \mathit{prove}(\mathit{lemma}(l_{d(j+1)}^{j+1})), l >, < \mathit{check}(\mathit{proof}(l_{d(j+1)}^{j+1})), c >;$
$< \mathit{prove}(\mathit{theorem\text{-}from}(L_{j+1})), t >; < \mathit{check}(\mathit{proof\text{-}from}(L_{j+1})), c >>.$

This plan is then reflected in a new collective commitment given by:

$$\begin{aligned}
&\text{C-COMM}_{G,Q}(\mathit{prove}(\mathit{theorem}(T))) \leftrightarrow \text{C-INT}_G(\mathit{prove}(\mathit{theorem}(T))) \wedge \\
&\quad \bigwedge_{i=1}^{d(j+1)} (\text{COMM}(l, t, \mathit{prove}(\mathit{lemma}(l_i^{j+1}))) \wedge \text{COMM}(c, l, \mathit{check}(\mathit{proof}(l_i^{j+1})))) \wedge \\
&\quad \text{COMM}(t, l, \mathit{prove}(\mathit{theorem\text{-}from}(L_{j+1}))) \wedge \text{COMM}(c, t, \mathit{check}(\mathit{proof\text{-}from}(L_{j+1}))) \wedge \\
&\quad \text{C-BEL}_G(\bigwedge_{\alpha \in Q} \bigvee_{a,b \in G} \text{COMM}(a, b, \alpha))
\end{aligned}$$

Thus, both l and t revise their individual intentions and social commitments towards the lemmas that are not in the new sequence L_{j+1}, whereas c adds individual intentions and social commitments to l and t with respect to their new proofs.

After each reconfiguration step, the agents continue executing their actions until a new block appears or the goal of proving the theorem is reached.

6 Discussion and conclusions

In this paper we have provided an abstract multi-level architecture which makes it possible to properly deal with DPS, based on the BDI paradigm. In particular, we stressed the need of the proper treatment of all motivational attitudes — individual, social, and collective, through all stages of DPS. To achieve this we introduced, as in [?], levels of abstraction reflecting the main aspects of DPS, namely potential recognition, team formation, plan generation, and plan execution. The main contribution is the reconfiguration algorithm respecting the property of continuity, which entails that replanning should be more efficient than when other (e.g. blind) methods are used. Also our definition of collective commitment is meant to ensure efficient replanning because of the pairwise social commitments involved. In addition, it seems that in many domains, conservativity is the appropriate criterion to guide local choices during reconfiguration.

[?] consider related issues, but do not provide a generic reconfiguration method and work with pre-given plans instead of plans that can be adapted on the fly. Also, their definition of collective intention is weaker than ours. In [?] an interesting definition of partial shared plans is given, but reconfiguration is not treated as a separate subject. [?] is based on the definitions of joint commitments and joint intentions of [?] and also has reconfiguration as its main subject matter. A strong point of that paper is the reconfiguration of complex joint team attitudes. The need for reconfiguration was recognized by other authors, and some of the above-mentioned papers treat certain aspects of reconfiguration or apply it to specific cases. Our paper, in contrast, provides a rather methodological approach maintaining all motivational attitudes.

When defining a general-purpose reconfiguration method some problems have been left aside. We do not take into account questions concerning heterogeneity of the system (see [?]). Our agents are allowed to be dissimilar and to have different problem solving perspectives, reflected in local expertise of domain solvers and global / strategic expertise of DPS organizers. However, all of them need to communicate, and some to cooperate and coordinate, despite their differences. We agree with [?] that agents' diversity should be restricted by possessing common ontologies, in this case with respect to action effects and motivational attitudes.

Future work will be to formalize the framework and to prove desired properties like efficiency of replanning. Next, the reconfiguration algorithm may be implemented for some applications.

Acknowledgments

We would like to thank professor Gerard Renardel de Lavalette for his helpful comments.
This work is supported by the Polish KBN Grant 8T11C 03110.