UDC 519.8:004.4, DOI:10.2298/CSIS100209016D

TeamLog in Action: a Case Study in Teamwork

Barbara Dunin-Kęplicz[1,2]*, Rineke Verbrugge[3]**, and Michał Ślizak[4]

[1] Institute of Informatics,
Warsaw University,
Banacha 2, 02-097 Warsaw, Poland
keplicz@mimuw.edu.pl
[2] Institute of Computer Science,
Polish Academy of Sciences,
Ordona 21, 01-237 Warsaw, Poland
Barbara.Dunin-Keplicz@ipipan.waw.pl
[3] Department of Artificial Intelligence,
University of Groningen,
P.O. Box 407, 9700 AK Groningen, The Netherlands
rineke@ai.rug.nl
[4] Institute of Informatics,
Warsaw University,
Banacha 2, 02-097 Warsaw, Poland,
Michal.Slizak@mimuw.edu.pl

Abstract. This article presents a case study of a theoretical multi-agent system designed to clean up ecological disasters. It focuses on the interactions within a heterogeneous team of agents, outlines their goals and plans, and establishes the necessary distribution of information and commitment throughout the team, including its sub-teams. These aspects of teamwork are presented in the TeamLog formalism [20], based on multi-modal logic, in which collective informational and motivational attitudes are first-class citizens. Complex team attitudes are justified to be necessary in the course of teamwork. The article shows how to make a bridge between theoretical foundations of TeamLog and an application and illustrates how to tune TeamLog to the case study by establishing sufficient, but still minimal levels for the team attitudes.

Keywords: TeamLog, motivational attitudes, multi-agent modelling.

1. Defining teamwork

When modeling complex automated systems it is often easier to think of them as a collection of agents, each capable of making its own decisions, rather than collections of modules with multiple, often unclear interdependencies. The notion of agency that is essential in the field of multi-agent systems (MAS) allows the designer to skip over the complexities of relationships between components,

* Supported by the MNiSW grant N N206 399334.
** Supported by the NWO Vici grant 016-094-603.

while still having the possibility to prove all required characteristics of the system, for example *liveness* and *correctness*. One of the most advanced models is the Belief-Desire-Intention (BDI) model of agency.

The BDI agent model comprises agents' individual beliefs, goals, and intentions. However, when a group of agents needs to work together in a planned and coherent way, agents' individual attitudes are not sufficient; the team needs to present a collective attitude over and above individual ones. Therefore, when constructing BDI systems, firstly a model of an agent as an *individual, autonomous* entity [35] has to be constructed. Then, a key point is to organize agents' cooperation in a way allowing the achievement of their common goal, while preserving, at least partly, their individual autonomy (see [31, 28, 42, 14, 16, 25, 34, 20] for some logical approaches to teamwork).

1.1. TEAMLOG **applied**

As a *static, descriptive* theory of collective motivational attitudes, TEAMLOG has been formed on the basis of individual goals, beliefs and intentions of strictly cooperative agents [14, 16, 22, 20]. It addresses the question what it means for a group of agents to have a *collective intention*, and then a *collective commitment* to achieve a common goal. While collective intention transforms a loosely-coupled group into a strictly cooperating team, collective commitment leads to team action, i.e., to coordinated realization of individual actions by committed agents according to a plan. The social plan can be constructed from first principles, or may be chosen from a repository of pre-constructed plans. Both collective intentions and collective commitments allow to fully express the potential of strictly cooperative teams [14, 16]. The bilateral and collective notions cannot be viewed as a sort of sum of individual ones.

In this case study we focus on *disaster response* and, more specifically, on decontamination of a certain polluted area. Our goal is to illustrate how to adjust a multi-agent system model created using the TEAMLOG theory to this specific use-case, with a focus on the theoretical performance of a potential implementation. Our contribution is to show how the costly infinitary definitions of collective attitudes can be reduced in a real-world situation, while maintaining the team cohesion necessary to achieve the goals. Whence the title "TEAMLOG in action" and the focus on full cooperation, also in a complex case, reflected in agents' individual, social (i.e. bilateral) and collective attitudes. Cases of competition are explicitly excluded by TEAMLOG definitions.

A theory of individual and group beliefs has been formalized in terms of epistemic logic [23, 32, 33]. General, common, and distributed knowledge and belief were defined in terms of agents' individual knowledge and belief. Different axiom systems express various properties of knowledge and belief, while the corresponding semantics naturally reflect these properties.

When modeling group attitudes, agents' awareness about the overall situation needs to be taken into account. Awareness is understood here as a limited form of consciousness: in a way typical for MAS, it refers to the state of an agent's beliefs about *itself* (*intra-personal*), about *others* (*inter-personal*) and about *the environment* (*group awareness*). Thus, various epistemic logics and different gradations of group information (from distributed belief to common knowledge) are adequate to formalize agents' awareness [23, 16, 33].

In TEAMLOG, group awareness is naturally expressed in terms of *common belief* (C-BEL_G), fully reflecting collective aspects of agents' behavior. Due to its infinitary flavor, this concept has a high complexity: its satisfiability problem is EXPTIME-complete [22]. There are general ways to reduce the complexity by restricting the language, by allowing only a small set of atomic propositions or restricting the modal context in formulas, as proved in [22, 21]. However, when building MAS applications, it may be more profitable to use domain-specific means to tailor TEAMLOG to the circumstances in question, calling for weaker forms of awareness [18]. In this case study we will show how to adjust team attitudes to ensure a lower complexity.

Collective intention constitutes a basis for preparing a plan, reflected finally in *collective commitment*. Various versions of collective commitments are applicable in different situations, dependent on organizational structure, communicative and observational possibilities and so on. These cause differences in agents' awareness both with respect to the *aspects* of teamwork of which they are aware, and the *kind* of awareness present within a team, ranging from distributed and individual belief, up to common knowledge. In this context, plan representation is a separate issue. It is assumed that any method of describing plans is acceptable, as long as it states the sequences of actions to be performed and allows us to project these actions onto individual agents. While a formal definition of social plan is available in TEAMLOG, the syntax used in the case study is considerably richer, and still very intuitive. It can easily be translated into combinations of definitions presented in [19, 15].

1.2. From real-world data to teamwork

Formal approaches to multi-agent systems are concerned with equipping software agents with functionalities for reasoning and acting. The starting point of most of them is the layer of beliefs, in the case of TEAMLOG extended by goals, intentions and commitments. These attitudes are usually represented in a symbolic, qualitative way. However, one should view this as an idealization. After all, agent attitudes originate from real-world data, gathered by a variety of sources at the *object level* of the system. Mostly, the data is derived from sensors responsible for perception, but also from hardware, different software platforms, and last but not least, from people observing their environment. The point is that this information is inherently quantitative. Therefore one deals with a meta-level duality: sensors provide quantitative characteristics, while reasoning tasks performed at the *meta-level* require the use of symbolic representations and inference mechanisms.

Research in the framework of TEAMLOG is structured along the lines depicted in Figure 1. The object-level information is assumed to be summarized in queries returning Boolean values. This way it is possible to abstract from a variety of formalisms and techniques applicable in the course of reasoning about

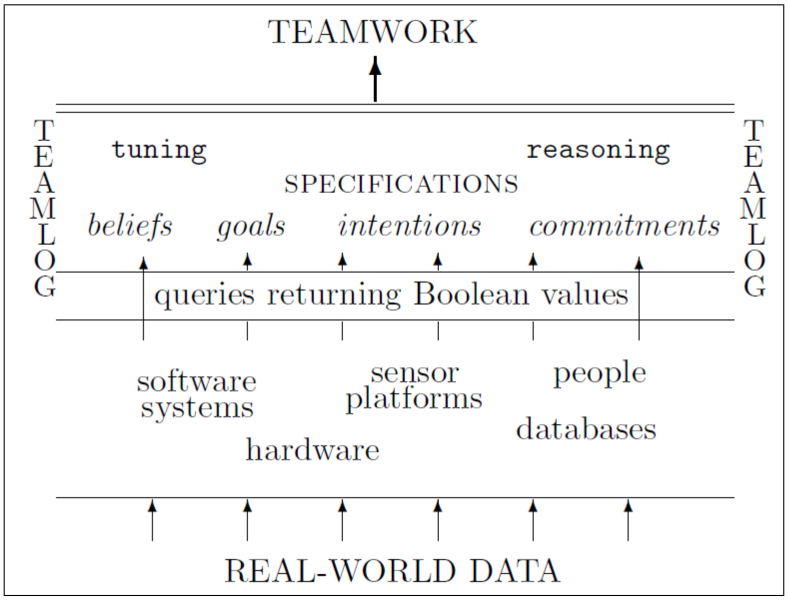

Fig. 1. The object- and meta-level view on teamwork

real-world data. This abstraction is essential, since the focus of TEAMLOG is on the meta-level, including formal specification and reasoning about teamwork.

This paper is structured into several sections. The first one introduces TEAMLOG and the problem in general terms. Next, in section 2, some definitions and assumptions regarding the environment are presented, including an outline of the interactions within a team and between sub-teams. This is followed in section 3 by definitions of social plans to be executed. In section 4 we take a closer look at the actions defined in our plan library and automated planning. In section 5 the TEAMLOG theory is presented. In section 6 we explore the minimal requirements for successful teamwork arising from the case study. Section 7 discusses related work, followed by a conclusion which sums up the article.

2. The case study: ecological disasters

Disaster management is a broad discipline related to dealing with and avoiding risks [41]. This involves several important tasks: *preparing for disaster* before it occurs, *disaster response* (e.g. emergency evacuation and decontamination), and *restoration* after natural or human-made disasters have occurred. In general, disaster management is the continuous process by which all individuals, groups, and communities manage hazards in an effort to avoid or ameliorate the impact of disasters resulting from them. Actions taken depend in part on perceptions of risk of those exposed [6]. Activities at each level (individual, group, community) affect the other levels.

The case study deals with response to ecological disasters caused by specific poisons, by means of heterogeneous multi-agent teams. The use-case is theoretical, in order to focus on those aspects of the problem that influence team action. The teams work in situations where time is critical and resources are scarce; it has been shown that teams consisting of software agents, robots and humans are a good choice in such critical situations [30, 37]. The example has been constructed by the authors, who took inspiration from the literature on disaster response [6, 41], heterogenous teams [30, 37], and unmanned aerial vehicles [8, 7].

Here, the main goal ($safe$) of teamwork is maintaining a given region REG safe or to return it to safety if it is in danger. The possible threats are two kinds of poison, X_1 and X_2, which are dangerous in high concentrations. The situation when both toxins mix is extremely hazardous, because if their concentration is high enough, they will react to form a chemical compound. This compound ($X_1 \oplus X_2$), once created, may explode if temperature and air pressure are high. Three functions f_1, f_2 and f_3 reflect the influence of temperature $t(A)$, air pressure $p(A)$ and concentrations $c_1(A)$ and $c_2(A)$ of poisons X_1 and X_2 at location A on the possible danger level at that location. The function ranges are divided into three intervals, as follows:

The first poison X_1:
- *safe*$_1$ iff $f_1(p(A), t(A), c_1(A)) \in [0, v_1]$;
- *risky*$_1$ iff $f_1(p(A), t(A), c_1(A)) \in (v_1, n_1]$;
- *dangerous*$_1$ iff $f_1(p(A), t(A), c_1(A)) \in (n_1, \infty)$;

The second poison X_2:
- *safe*$_2$ iff $f_2(p(A), t(A), c_2(A)) \in [0, v_2]$;
- *risky*$_2$ iff $f_2(p(A), t(A), c_2(A)) \in (v_2, n_2]$;
- *dangerous*$_2$ iff $f_2(p(A), t(A), c_2(A)) \in (n_2, \infty)$;

The chemical compound $X_1 \oplus X_2$:
- *safe*$_3$ iff $f_3(p(A), t(A), c_1(A), c_2(A)) \in [0, v_3]$;
- *risky*$_3$ iff $f_3(p(A), t(A), c_1(A), c_2(A)) \in (v_3, n_3]$;
- *explosive* iff $f_3(p(A), t(A), c_1(A), c_2(A)) \in (n_3, \infty)$;

The proposition $safe$ is defined as $safe :=$ *safe*$_1 \wedge$ *safe*$_2 \wedge$ *safe*$_3$ and is also referred to as a goal. There are also thresholds ϵ_1 and ϵ_2: when the concentration of a poison X_i exceeds ϵ_i, the respective function f_i is computed.

2.1. Starting point: the agents

This model reflects cooperation between humans, software agents, robots, unmanned aerial vehicles ($UAVs$) as discussed in [8, 7], and a helicopter steered by a pilot.

The whole process is controlled by one $coordinator$, who initiates cooperation, coordinates teamwork between different sub-teams of the full team G, is responsible for dividing the disaster zone into sectors and assigning a sub-team to each sector to perform clean-up. Several sub-teams $G_1, \ldots G_k \subseteq G$ of similar make-up work in parallel, aiming to prevent or neutralize a contamination. Each of these sub-teams G_i consists of:

- one UAV_i - responsible to the coordinator for keeping assigned sectors safe. This agent cannot carry a heavy load, but it carries a computer, therefore it has considerable computational capabilities for planning and is capable of observing and mapping the terrain;
- n_i identical neutralizing robots $rob_{i_1}, \ldots, rob_{i_{n_i}}$ - responsible to their UAV_i for cleaning up a zone.

Next to the sub-teams there is a rather independent member of G:

- one regular helicopter steered by the human $pilot$, who can independently choose the order of cleaning up assigned areas, is directly accountable to the coordinator and can communicate as an equal with the $UAVs$.

See Figure 2 for the team structure.

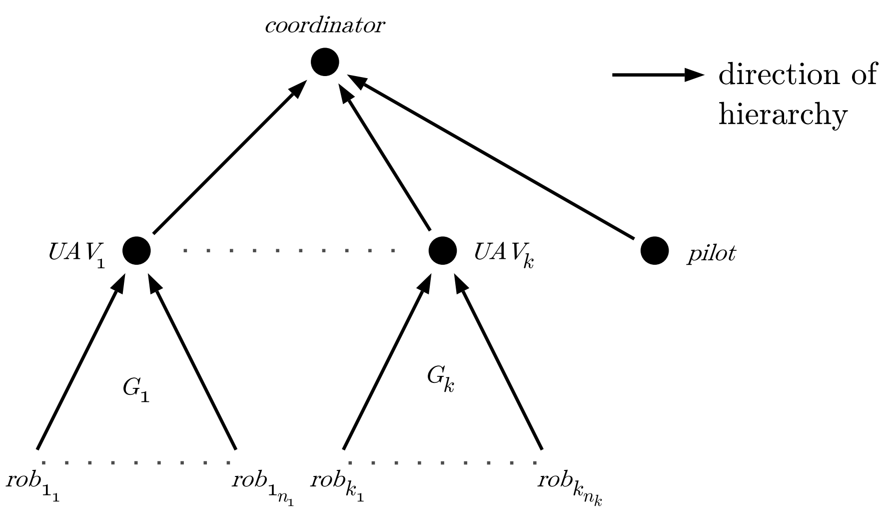

Fig. 2. Hierarchical team structure of the ecological disaster management team G.

2.2. Cooperation between sub-teams

The entire disaster zone is divided into sectors by the coordinator, based on terrain type, sub-team size, population density and hot spots known in advance (see Figure 3). Sub-teams are responsible for (possibly many) sectors. The leader UAV_i of a sub-team G_i prepares a plan to keep its sectors safe. Each plan is judged based on a fitting function fit, which takes into account:

1. available robots, including their current task, load, capacity and position;
2. whether the plan relies on the help from other sub-teams;
3. task priorities;
4. the minimum amount of time it takes to implement;
5. the minimum number of robots it requires.

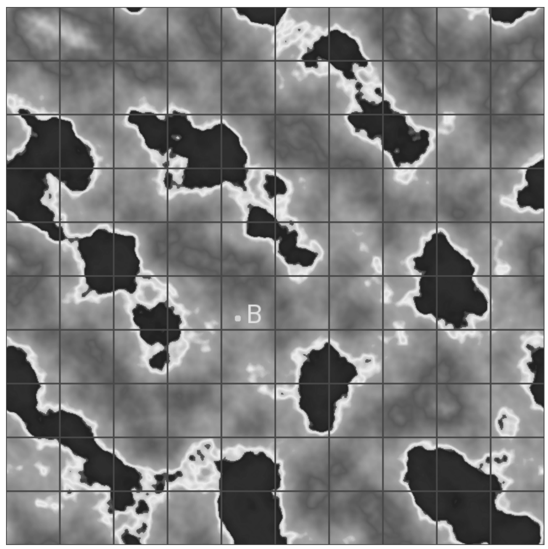

Fig. 3. Example terrain split, with base for robots B marked.
The darkest regions consist of water and are an obstacle to the cleanup robots.

The $UAVs$ communicate and cooperate with one another. If performing tasks requires more robots than are currently available in their own sub-team G_i, its leader UAV_i can call for reinforcements from another UAV_j, for $j \leq k, j \neq i$. Of course for UAV_j in question, fulfilling its own sub-team G_j's objectives has priority over helping others from G_i.

2.3. A bird's-eye view on cases

To maintain the goal $safe$, the situation is monitored by the coordinator and the $UAVs$ on a regular basis with frequency *freq*. During *situation recognition*, in the risky cases monitoring is performed twice as frequently. Depending on the mixture and density of poisons in a location, some general cases followed by the relevant procedures are established. All remedial actions are to be performed relative to the contaminated area:

Case $safe$**:** True $\longrightarrow$ *situation recognition*

Case *dangerous*$_\mathbf{1}$**:** *rain* $\longrightarrow$ *liquid* L_1 *to be poured on the soil*
normal or dry $\longrightarrow$ *liquid* L_2 *to be sprayed from the air*

Case *dangerous*$_\mathbf{2}$**:** *rain* $\longrightarrow$ *solid* S_1 *to be spread*, followed by *liquid catalyst* K_1 *to be poured*
normal or dry $\longrightarrow$ *solid* S_1 *to be spread*

Case *explosive***:** *before explosion* $\longrightarrow$ *evacuation*
after explosion $\longrightarrow$ *rescue action*

3. Global plans

While team planning is a very interesting problem, the algorithms for solving it are exponential for most cases. For performance reasons it was decided to use a pre-compiled plan library (consisting of plans devised by us) in the case study. Please observe that while the existence of such a library simplifies planning, it doesn't eliminate the need for it. Some actions (e.g. $move$) presented in the plan library are complex, and can only be solved with specialized planning techniques.

The ad-hoc notation representing high-level plans can easily be translated into basic definitions introduced in [19, 15]. The only requirement in TEAMLOG is that the plan describes the order of actions and their performers. Furthermore, TEAMLOG doesn't enforce any representation of time in plans. A sensible implementation of a planner designed to work in a physical environment will have to synchronize agents' actions, but the specific representation of time is not an issue at the level of abstraction used here. Clearly, the planner will be able to obtain temporal dependencies between actions from the presented description.

In order to control the amount of interactions and decrease the time needed to establish beliefs, we introduce a hierarchical team model. The coordinator views a sub-team G_i as a single entity, even though the $UAVs$ manage the work of many autonomous neutralizing robots. All agents are initially located at a base B, which also houses chemicals required for decontamination. It may interact with agents via communication protocols that would guarantee that agents do not compete for resources such as decontaminants.

3.1. The social plan $\langle Cleanup \rangle$

The global social plan for which the coordinator and $UAVs$ are responsible, is designed with respect to location A. It is a while-loop, in which observation is interleaved with treatment of current dangers by level of priority, from most to least dangerous. The goal (denoted as $Clean$) is to keep locations in a $safe$ state. All subplans mentioned in $\langle Cleanup \rangle$, namely $\langle$ Plan $SR\rangle$, $\langle$ Plan E $\rangle$,

$\langle$ Plan D_1R $\rangle$, $\langle$ Plan D_1N $\rangle$, $\langle$ Plan D_2R $\rangle$, and $\langle$ Plan D_2N $\rangle$ are described more precisely in the subsequent subsections.

begin
  freq $:= a$;
  $\{freq$ - interval between two checks of the environment$\}$
  while true do
    $\langle$ Plan $SR\rangle$ $\{$Assess the situation at A, with frequency *freq*$\}$
    if *explosive* then do $\langle$ Plan E $\rangle$ end;
    elif *dangerous*$_1$ and rain then do $\langle$ Plan D_1R $\rangle$ end;
    elif *dangerous*$_1$ then do $\langle$ Plan D_1N $\rangle$ end;
    elif *dangerous*$_2$ and rain then do $\langle$ Plan D_2R $\rangle$ end;
    elif *dangerous*$_2$ then do $\langle$ Plan D_2N $\rangle$ end;
    elif *risky*$_1$ $\vee$ *risky*$_2$ $\vee$ *risky*$_3$ then *freq* $:= \frac{a}{2}$ end
    else $\{$safe situation$\}$ *freq* $:= a$ end;
  end
end.

Here, a represents the default frequency of checking the environment. This interval is shortened when a risky situation is encountered, to handle such a case with more caution. Dividing the interval by 2 is done to make the example specific; it is easy to adapt the plan to frequency adjustments applicable in a real-world scenario.

3.2. The social plan $\langle SR\rangle$

This plan performs situation recognition at location A. One of the $UAVs$, for example UAV_1, is responsible for monitoring. Alternatively, this task could be assigned as a joint responsibility to $UAV_1, \ldots, UAV_k$. However, that solution would require information fusion which is in general a very complex process.

begin
  $C_1 := c_1(A)$ $\{C_1$ is the measured concentration of poison X_1 at $A\}$
  $C_2 := c_2(A)$ $\{C_2$ is the measured concentration of poison X_2 at $A\}$
  $T := t(A)$ $\{T$ is the measured temperature at $A\}$
  $P := p(A)$ $\{P$ is the measured air pressure at $A\}$
  $\{$Computation of the situation at $A\}$
  if $C_1 > \epsilon_1$ then compute $f_1(C_1, T, P)$ end;
  if $C_2 > \epsilon_2$ then compute $f_2(C_2, T, P)$ end;
  if $C_1 > \epsilon_1$ and $C_2 > \epsilon_2$ then compute $f_3(C_1, C_2, T, P)$ end;
end.

3.3. The social plan $\langle E \rangle$

After an explosion, evacuation and rescue of people should take place. This subject is discussed in many studies [6, 41, 30] and will not be elaborated here. Instead, other subplans included in $\langle Cleanup \rangle$ are presented.

3.4. The social plan $\langle D_1R \rangle$

This plan is applicable when *dangerous*$_1$ occurs under weather condition *rain*. Each UAV_i may be allocated this social plan for a given location as decided by the *coordinator*.
Goal $\psi_1(L_1)$: to apply liquid L_1 on all areas contaminated with poison X_1.

$\{$Assumption: One portion of L_1 neutralizes poison X_1 at a single location.$\}$

while $contaminated\text{-}area \neq emptyset$ do
begin
  $A := calculate(UAV_i, \{rob_{i_j}\})$;
  $\{UAV_i$ finds region A for rob_{i_j} to clean up$\}$
  $get(rob_{i_j}$, L_1, B); $\{rob_{i_j}$ retrieves a tank with liquid L_1 from location $B\}$
  $path := get\_path(UAV_i, rob_{i_j}, B, A)$; $\{rob_{i_j}$ requests a path to follow$\}$
  $move(rob_{i_j}, path)$; $\{rob_{i_j}$ moves from location B to location $A\}$
  $pour(rob_{i_j}, L_1, A)$;
  $contaminated\text{-}area := contaminated\text{-}area \setminus A$;
  $return\_path := get\_path(UAV_i, rob_{i_j}, A, B)$;
  $move(rob_{i_j}, return\_path)$;
end.

Here and in subsequent social plans, the assumption about needing a single portion of a liquid (or of solid and catalyst, respectively) per location is meant as an illustrative example. Functions computing the needed amounts of these substances from the measured concentrations of poisons could be used in a real-world case. It is also assumed that UAV_i establishes priorities of locations to be cleaned when planning the robots' routes, taking into account that some robots should start clearing certain paths such that other robots can then safely traverse these.

3.5. The social plan $\langle D_1N \rangle$

This plan is applicable when *dangerous*$_1$ occurs under weather condition *normal or dry*. The spraying is usually performed by the pilot on request from one of the $UAVs$. In the plan below, UAV stands for any of $UAV_1, \ldots, UAV_k$.
Goal $\psi_2(L_2)$: to spray liquid L_2 on areas contaminated with poison X_1.

$\{$Assumption: One portion of L_2 neutralizes poison X_1 at a single location.$\}$
$\{$Assumption: The helicopter can transport k portions of liquid L_2.$\}$

while $contaminated\text{-}area \neq emptyset$ do
begin
$request(UAV, coordinator, pilot, \psi(L_2))$;
$confirm(pilot, UAV, coordinator, \psi(L_2))$;
$request(pilot, UAV, list_1, k)$;
$send(UAV, pilot, list_1)$; {$list_1$ has at most k contaminated areas}
$upload(helicopter, L_2, |list_1|)$; {$pilot$ retrieves required amount of liquid L_2}
$take\text{-}off(helicopter, B)$; {$pilot$ takes off from location B}
do $\langle plan\text{-}for\text{-}spraying(helicopter, L_2, list_1)\rangle$;
{$pilot$ sprays L_2 using his own invented plan}
$confirm(pilot, UAV, done(plan\text{-}for\text{-}spraying(helicopter, L_2, list_1))$;
$contaminated\text{-}area := contaminated\text{-}area \setminus list_1$;
$landing(helicopter, B)$;
$free(pilot, coordinator)$;
end.

3.6. The social plan $\langle D_2R\rangle$

This plan is applicable when *dangerous*$_2$ occurs under weather condition *rain*. Goal $\psi_3(S_1, K_1)$: to spread solid S_1 on all areas contaminated with poison X_2, followed by applying catalyst K_1 to all areas where S_1 is present.

{Assumption: One portion of S_1 and K_1 neutralize poison X_2 at a single location.}

while $contaminated\text{-}area \neq emptyset$ do
begin
$A := calculate(UAV_i, \{rob_{i_j}, rob_{i_l}\})$;
{UAV_i finds region for rob_{i_j} and rob_{i_l} to spread solid and catalyst, respectively.}

begin_parallel {two main operations are done in parallel: applying a solid to the area, and pouring a catalyst on it }
{a plan similar to $\langle D_1R\rangle$, but using S_1:}
$get(rob_{i_j}, S_1, B)$; {rob_{i_j} retrieves a portion of solid S_1 from location B}
$path := get\_path(UAV_i, rob_{i_j}, B, A)$; {$rob_{i_j}$ requests a path to follow}
$move(rob_{i_j}, path)$; {rob_{i_j} moves from location B to location A}
$pour(rob_{i_j}, S_1, A)$;
$return\_path := get\_path(UAV_i, rob_{i_j}, A, B)$;
$move(rob_{i_j}, return\_path)$;
||
$get(rob_{i_l}, K_1, B)$;
$path := get\_path(UAV_i, rob_{i_l}, B, A)$;
$move(rob_{i_l}, path)$;
$wait\_for(spread(S_1, A))$; {rob_{i_l} waits for *someone* to spread S_1 in A}
$pour(rob_{i_l}, K_1, A)$;

$return\_path := get\_path(UAV, rob_{i_l}, A, B)$;
$move(rob_{i_l}, return\_path)$;
end_parallel
$contaminated\text{-}area := contaminated\text{-}area \setminus A$;
end.

The planner executed by UAV_i computes the paths in such a way that the robots' movements are synchronized, so that they arrive in area A almost simultaneously.

3.7. Plan $\langle D_2N \rangle$

This plan, very similar to $\langle D_1R \rangle$, is applicable when *dangerous*$_2$ occurs under weather condition *normal or dry*. Each UAV_i may be allocated $\langle D_2N \rangle$ for a given location as decided by the *coordinator*. Goal $\psi_1(S_1)$ is to apply solid S_1 on all areas contaminated with poison X_2.

{Assumption: One portion of S_1 neutralizes poison X_2 at a single location.}

while $contaminated\text{-}area \neq emptyset$ do
begin
$A := calculate(UAV_i, \{rob_{i_j}\})$; {$UAV_i$ finds region A for rob_{i_j} to clean up}
$get(rob_{i_j}$, S_1, B); {rob_{i_j} retrieves a portion of solid S_1 from location B}
$path := get\_path(UAV_i, rob_{i_j}, B, A)$; {$rob_{i_j}$ requests a path to follow}
$move(rob_{i_j}, path)$; {rob_{i_j} moves from location B to location A}
$pour(rob_{i_j}, S_1, A)$;
$contaminated\text{-}area := contaminated\text{-}area \setminus A$;
$return\_path := get\_path(UAV_i, rob_{i_j}, A, B)$;
$move(rob_{i_j}, return\_path)$;
end.

The decontamination action is usually successful. Unfortunately, its effects may not be immediately visible, therefore the plans don't check contamination levels right after applying chemicals.

Even though $\langle D_1R \rangle$, $\langle D_1N \rangle$, $\langle D_2R \rangle$, and $\langle D_2N \rangle$ do not contain explicit actions to check the success of decontamination, such checks are part and parcel of the main $\langle Cleanup \rangle$ plan, so the coordinator and $UAVs$ can re-plan accordingly if decontamination does not succeed.

4. Defining actions

The actions are defined using the standard STRIPS syntax and semantics:

1. For every $i \in G$, A_i is the set of actions that agent i is capable of performing;
2. Each action $a \in A = \bigcup_{i \in G} A_i$ is a tuple $\langle \alpha, \beta, \gamma, \delta \rangle$, where:

α is the set of all conditions that must be true for the action to be executable;
β is the set of all conditions that must be false for the action to be executable;
γ is the set of all conditions that become true as a result of this action;
δ is the set of all conditions that become false as a result of this action.

4.1. Actions in the case study

Some actions are complex and require further planning. Definitions of the robots' actions are as follows:

- $get(Ag, Ob, From) =$
  $\langle\{at(Ag, From), at(Ob, From), available(Ob), capable\_of\_lifting(Ag, Ob)\},$
  $\emptyset, \{has(Ag, Ob)\}, \{available(Ob), at(Ob, From)\}\rangle$
- $move(Ag, From, Direction) =$
  $\langle\{at(Ag, From), in\_direction(From, Direction, X), can\_enter(Ag, X)\},$
  $\emptyset, \{at(Ag, X)\}, \{at(Ag, From)\}\rangle$
- $wait\_for(Cond) = \langle\emptyset, \emptyset, \{Cond\}, \emptyset\rangle$
- $pour(Ag, Chem, Loc) =$
  $\langle\{at(Ag, Loc), contains(Ob, Chem), has(Ag, Ob)\}, \emptyset, \emptyset, \{has(Ag, Ob)\}\rangle$

The complex action $move(Ag, From, To)$ results from planning performed by agent Ag (it might request terrain information from its controlling UAV in the process). It will be implemented as a series of atomic actions of the form $move(Ag, From, Direction)$.

Definitions of the pilot's actions:

- $upload(helicopter, Chem, Num) =$
  $\langle\{at(helicopter, B)\},$
  $\{airborne(helicopter)\}, \{has(helicopter, Chem, Num)\}, \emptyset\rangle$
- $take\text{-}off(helicopter, Loc) =$
  $\langle\{at(helicopter, Loc)\},$
  $\{airborne(helicopter)\}, \{airborne(helicopter)\}, \emptyset\rangle$
- $landing(helicopter, Loc) =$
  $\langle\{at(helicopter, Loc), airborne(helicopter)\},$
  $\emptyset, \emptyset, \{airborne(helicopter)\}\rangle$
- $plan\text{-}for\text{-}spraying(helicopter, Chem, \mathit{list}_1) =$
  $\langle\{has(helicopter, Chem, X), X \geq |\mathit{list}_1|,$
  $capable\_of\_lifting(helicopter, Chem, |\mathit{list}_1|), airborne(helicopter)\},$
  $\emptyset, \{has(helicopter, Chem, X)\}, \{has(helicopter, Chem, Y), Y \equiv X - |\mathit{list}_1|\}\rangle$

Here $plan\text{-}for\text{-}spraying(helicopter, Chem, \mathit{list}_1)$ is considered an atomic action by the planner, as the pilot is an independent agent planning for himself. Notation $|\mathit{list}_1|$ stands for the length of list_1.

4.2. Plans and plan projections

A *social plan* is defined as a composition of simple social plan expressions.

SP1 $\langle a, i \rangle$ is a (simple) social plan expression iff $i \in G \wedge a \in A_i$
SP2 if α and β are social plan expressions and φ is any true/false statement then:
- $\langle \alpha; \beta \rangle$ is a (compound) social plan expression (sequential execution)
- $\langle \alpha || \beta \rangle$ is a (compound) social plan expression (parallel execution)
- $\langle$ if φ then do α end $\rangle$ is a (compound) social plan expression (an if-then statement)

The notation ξ is used to refer to an empty plan:

- $\langle \alpha; \xi \rangle \equiv \alpha$
- $\langle \xi; \beta \rangle \equiv \beta$
- $\langle \alpha || \xi \rangle \equiv \alpha$
- $\langle \xi || \beta \rangle \equiv \beta$
- $\langle$ if φ then do ξ end $\rangle \equiv \xi$

A plan is written for roles to be adopted by agents. Each role has requirements assuring, for example, that a robot cannot assume the role of a pilot (since it is not capable of flying a helicopter). A plan projection for agent i can be understood as the individual view of the plan, where some of the roles have been adopted by agent i. An *agent i's projection of a social plan P* (denoted $P|_i$) is defined as:

- $\langle a, i \rangle|_j \equiv \langle a, i \rangle$ iff $i = j$
- $\langle a, i \rangle|_j \equiv \xi$ iff $i \neq j$
- $\langle \alpha; \beta \rangle|_i \equiv \langle \alpha|_i; \beta|_i \rangle$
- $\langle \alpha || \beta \rangle|_i \equiv \langle \alpha|_i || \beta|_i \rangle$
- $\langle$ if φ then do α end $\rangle|_i \equiv \langle$ if φ then do $\alpha|_i$ end $\rangle$

A projection of a social plan P for a group G of agents can be defined in a similar fashion, using $i \in G$ instead of $i = j$.

4.3. Plan projections in the case study

Agents naturally commit to their controlling UAV which acts as a 'middle manager' on behalf of the *coordinator*. Each UAV is committed to the *coordinator* with regard to the task of keeping all assigned regions in a $safe$ state.

Each agent has its own projection of the overall plan.

- The *coordinator* is aware of the $\langle Cleanup \rangle$ plan in the context of all regions;
- $UAVs$ need a projection of the $\langle Cleanup \rangle$ plan in all areas to which they are assigned;
- Robots need to have a projection of the $\langle Cleanup \rangle$ plan only regarding actions which they may take part in.

Projections for the pilot are not considered here, since he works independently.

5. TEAMLOG: a logical theory for teamwork

Due to the space limit, only the relevant fragment of TEAMLOG will be presented here. For extensive explanation and discussion see [14–16, 20].

5.1. Beliefs in TEAMLOG

For the individual part, a standard $KD45_n$ system for n agents has been adopted, governing the individual belief operator BEL, as explained in [23]. Additionally, for group beliefs, with a group $G \subseteq \{1, \ldots, n\}$, as in [23]:

C1 $\text{E-BEL}_G(\varphi) \leftrightarrow \bigwedge_{i \in G} BEL(i, \varphi)$ *(general belief: "everyone believes")*

More general iterated form for $k \geq 2$:

$\text{E-BEL}^k_G(\varphi) \leftrightarrow \text{E-BEL}^{k-1}_G(\text{E-BEL}_G(\varphi))$, where $\text{E-BEL}^1_G(\varphi) \equiv \text{E-BEL}_G(\varphi)$

A very strong and heavily used notion in TEAMLOG is that of *common belief*: $\text{C-BEL}_G(\varphi)$ - "it is common belief in the group G that φ is true":

C2 $\text{C-BEL}_G(\varphi) \leftrightarrow \text{E-BEL}_G(\varphi \wedge \text{C-BEL}_G(\varphi))$
RC1 From $\varphi \to \text{E-BEL}_G(\psi \wedge \varphi)$ infer $\varphi \to \text{C-BEL}_G(\psi)$ *(induction)*

Common belief is a very powerful notion. For example, if $\text{C-BEL}_G(\varphi)$ holds then $\text{C-BEL}_G(\psi)$ holds as well with ψ being any logical consequence of φ. This ensures that agents reach the same conclusions from φ and commonly believe in them.

5.2. Tuning collective attitudes

Collective notions from TEAMLOG ensure the calibration of agents' *awareness*. In this case study the strength of collective attitudes is adjusted to a specific domain, while group attitudes are set at the minimal level ensuring effective team operation. In general, the question regarding the level of awareness about each specific aspect of teamwork needs to be addressed.

Instances of $awareness_G$ in TEAMLOG definitions can be anything from $\emptyset$, through individual beliefs, different levels of E-BEL^k_G, to common belief C-BEL_G. Stronger levels of belief require increased communication to ensure their proper propagation. It has been argued that in ideal teamwork, $awareness_G$ is taken to be C-BEL_G [14]. Assuming that the communication medium is perfect, it is possible to attain this. In practical implementations in an asynchronous, uncertain medium, common knowledge (C-KNOW_G) has been proven to be impossible to achieve [29], and common belief (C-BEL_G) only under extremely restricted constraints [17]. Alternatively, in realistic circumstances, one can apply communication protocols that establish ever higher approximations of common belief within a group [4, 2]. These protocols are less efficient because the required number of messages passed back and forth between the initiator and other agents grows linearly with the desired level of iteration. However, the underlying assumptions are much easier to guarantee.

5.3. Intentions in TEAMLOG

For the individual intention INT, the TEAMLOG axioms comprise the system KD_n, including the intention consistency axiom D. In addition, the system developer may choose whether to add negative introspection of intentions (see [14]).

It is certainly not sufficient that all members of the team G have the associated individual intention $\mathrm{INT}(i, \varphi)$ to achieve φ, i.e. a *general intention*. To exclude competition, all agents should *intend* all members to have the associated individual intention, as well as the intention that all members have the individual intention, and so on; this is called a *mutual intention* ($\mathrm{M\text{-}INT}_G(\varphi)$). Furthermore, all team members are aware of this mutual intention: $awareness_G(\mathrm{M\text{-}INT}_G(\varphi))$. Please note that team members remain autonomous in maintaining their other motivational attitudes, and may compete about other issues.

M1 $\mathrm{E\text{-}INT}_G(\varphi) \leftrightarrow \bigwedge_{i \in G} \mathrm{INT}(i, \varphi)$ *(general intention: "everyone intends")*

$\mathrm{E\text{-}INT}^k_G$ is also defined iteratively, similarly to higher-order general beliefs, for $k \geq 2$:

M1′ $\mathrm{E\text{-}INT}^k_G(\varphi) \leftrightarrow \mathrm{E\text{-}INT}^{k-1}_G(\mathrm{E\text{-}INT}_G(\varphi))$, where $\mathrm{E\text{-}INT}^1_G(\varphi) \equiv \mathrm{E\text{-}INT}_G(\varphi)$

Mutual and collective intentions are governed by:

M2 $\mathrm{M\text{-}INT}_G(\varphi) \leftrightarrow \mathrm{E\text{-}INT}_G(\varphi \wedge \mathrm{M\text{-}INT}_G(\varphi))$ *(mutual intention)*
RM1 From $\varphi \to \mathrm{E\text{-}INT}_G(\psi \wedge \varphi)$ infer $\varphi \to \mathrm{M\text{-}INT}_G(\psi)$ *(induction)*
M3 $\mathrm{C\text{-}INT}_G(\varphi) \leftrightarrow \mathrm{M\text{-}INT}_G(\varphi) \wedge awareness_G(\mathrm{M\text{-}INT}_G(\varphi))$ *(collective intention)*

5.4. Collective commitment in TEAMLOG

After a group is constituted via collective intention, a *collective commitment* between the team members needs to be established. While a collective intention is an inspiration for team activity, the plan-based collective commitment expresses the case-specific details provided by planning and action allocation. It is reflected in bilateral commitments towards individual actions. A bilateral commitment from agent i towards agent j to perform action α is represented as $\mathrm{COMM}(i, j, \alpha)$. In this article, bilateral commitment is viewed as a primitive notion, but see [16] for its characterization and governing axiom.

5.5. Collective commitment schema

A flexible tuning schema for *collective commitments* is presented in [16]. In summary, group G has a *collective commitment* to achieve goal φ based on social plan P ($\mathrm{C\text{-}COMM}_{G,P}(\varphi)$) iff all of the following hold (in the corresponding definition below, parts between curly brackets may or may not be present): The group mutually intends φ (with or without being aware of this); moreover, successful execution of social plan P leads to φ ($cons(\varphi, P)$) (again, with or without the group being aware of this); and finally, for every action α from plan P, there is one agent in the group who is bilaterally committed to another agent in the group to fulfil that action ($\mathrm{COMM}(i, j, \alpha)$) (with or without the group being aware of this):

$$\begin{aligned}&\text{C-COMM}_{G,P}(\varphi) \leftrightarrow \text{M-INT}_G(\varphi) \wedge \{awareness_G(\text{M-INT}_G(\varphi))\} \wedge \\ &\quad cons(\varphi, P) \wedge \{awareness_G(cons(\varphi, P))\} \wedge \\ &\quad \textstyle\bigwedge_{\alpha \in P} \bigvee_{i,j \in G} \text{COMM}(i, j, \alpha) \wedge \{awareness_G(\bigwedge_{\alpha \in P} \bigvee_{i,j \in G} \text{COMM}(i, j, \alpha))\}\end{aligned}$$

Strong collective commitment Different types of collective commitments related to different organizational structures and environments have been introduced in terms of a 'tuning machine' [16]. One instantiation of the above scheme is the *strong collective commitment* [16]. In this case, the team knows the overall goal, collectively believes that the social plan is correct ($\text{C-BEL}_G(cons(\varphi, P))$) and that things are under control. However, team members do not need to know exactly who is responsible for each task.

Strong collective commitment is applicable in teams with no collective responsibility:

$$\begin{aligned}&\text{S-COMM}_{G,P}(\varphi) \leftrightarrow \text{C-INT}_G(\varphi) \wedge cons(\varphi, P) \wedge \\ &\quad awareness_G(cons(\varphi, P)) \wedge \textstyle\bigwedge_{\alpha \in P} \bigvee_{i,j \in G} \text{COMM}(i, j, \alpha) \wedge \\ &\quad awareness_G(\textstyle\bigwedge_{\alpha \in P} \bigvee_{i,j \in G} \text{COMM}(i, j, \alpha))\end{aligned}$$

Weak collective commitment In *weak collective commitment* [16], the team knows the overall goal, but doesn't know the details of the plan: there is no collective awareness of the plan's correctness, even though actions have been appropriately allocated.

Weak collective commitment is applicable in teams with a dedicated planner, who takes care of the proper planning reflected in $cons(\varphi, P)$:

$$\begin{aligned}&\text{W-COMM}_{G,P}(\varphi) \leftrightarrow \text{C-INT}_G(\varphi) \wedge cons(\varphi, P) \wedge \\ &\quad \textstyle\bigwedge_{\alpha \in P} \bigvee_{i,j \in G} \text{COMM}(i, j, \alpha) \wedge awareness_G(\bigwedge_{\alpha \in P} \bigvee_{i,j \in G} \text{COMM}(i, j, \alpha))\end{aligned}$$

Team commitment A further weakening of the group commitment results in the *team commitment* [16]. This case differs from *weak collective commitment* including solely obligatory elements of collective commitment, without the team being aware of them. Therefore, agents can't infer if any action they take will result in completion of plan P.

Team commitment is applicable in teams where agents only receive orders and don't help each other unless specifically told what to do.

$$\text{T-COMM}_{G,P}(\varphi) \leftrightarrow \text{C-INT}_G(\varphi) \wedge cons(\varphi, P) \wedge \textstyle\bigwedge_{\alpha \in P} \bigvee_{i,j \in G} \text{COMM}(i, j, \alpha)$$

6. Adjusting the TEAMLOG notions to the case study

6.1. C-COMM_G on the sub-team level

Within a sub-team G_i, the UAV_i has the highest level of awareness as it knows the entire $\langle Cleanup \rangle$ plan for a particular region. There is no need for others to know the details of that plan.

The strength of the *collective commitment* depends on the assumptions regarding cooperation between cleanup robots. There are two possible scenarios:

– Robots help one another only when their UAV orders them to.
  In this case, robots from G_i need a quite limited awareness of the plan. For example, they need to know the partially instantiated subplans applicable in dangerous situations ($\langle D_1R\rangle$, $\langle D_2R\rangle$ or $\langle D_2N\rangle$). In the relevant weather conditions, they may need to carry out one of these subplans for a region assigned by the UAV_i. This UAV_i also assigns roles to robots, who do not know the shares of others. Thus, *team commitment* is sufficient on the sub-team level.
– Robots help each other voluntarily.
  In this case, they will also need to be aware about the partially instantiated plans of nearby robots from G_i in order to assist and assume a role that one of its colleagues fails to perform. With regard to the $\langle Cleanup\rangle$ plan, this corresponds to *weak collective commitment* on the sub-team level. In addition, it requires $\bigwedge_{\alpha\in P}\bigvee_{m,n\in G_i}\text{E-BEL}_{G_i}(\text{COMM}(m,n,\alpha))$. That specific belief is necessary if agents are to pitch in for one another.

6.2. C-COMM_G on the team level

In team G the *coordinator* has the highest awareness; for example, it is in charge of the global planning, as explained in section 2.2. The $UAVs$ in its team only need to know their projection of the overall plan, and believe that the entire plan has been shared among them. The $coordinator$ knows both the plan and action allocation. The *coordinator*, the *pilot* and and all $UAVs$ are aware that $cons(safe, Cleanup)$ holds.

With regard to the $\langle Cleanup\rangle$ plan, this corresponds to *strong collective commitment* on the top team level, where all $UAVs$ and the pilot make social commitments to the coordinator. Taking $G' = \{coordinator, pilot, UAV_1, \ldots, UAV_k\}$ and putting the $awareness_{G'}$ dial at common belief, we have:

$$
\begin{aligned}
&\text{S-COMM}_{G',Cleanup}(safe) \leftrightarrow \text{C-INT}_{G'}(safe) \wedge cons(safe, Cleanup) \wedge \\
&\quad \text{C-BEL}_{G'}(cons(safe, Cleanup)) \wedge \\
&\quad \textstyle\bigwedge_{\alpha\in Cleanup}\bigvee_{i\in G'}\text{COMM}(i, coordinator, \alpha) \wedge \\
&\quad \text{C-BEL}_{G'}(\textstyle\bigwedge_{\alpha\in Cleanup}\bigvee_{i\in G'}\text{COMM}(i, coordinator, \alpha))
\end{aligned}
$$

The dialogues taking place between the coordinator and its direct subordinates during the creation of collective commitments ensure that the necessary awareness is established (see [20, Chapter 8]).

6.3. Organization structure: who is socially committed to whom?

Commitments in the team follow the organizational structure of Figure 2. The coordinator is socially committed to achieving the overall goal, with respect to the main social plan. Other agents are committed to their projection of that plan. The coordinator is committed towards itself and towards the relevant control authority, for example, the national environmental agency for which it works.

Each UAV_i for $i = 1, \ldots, k$ is committed towards the coordinator with respect to achieving its part of the plan, namely keeping specified regions safe.

The robots in G_i for $i = 1, \ldots, k$ commit to perform their share to their leading UAV_i, which has the power to uncommit them. There is a clear hierarchy where the coordinator is the leader of the team G as a whole, while the $UAVs$ are 'middle-managers' of sub-teams. The $UAVs$ also sometimes commit to a colleague UAV when some of their robots are temporarily delegated to the other's sub-team.

The human pilot has a somewhat special role in that he does not manage any sub-team. Instead, he directly commits to the coordinator, or sometimes to $UAVs$ if they request his assistance.

6.4. Minimal levels of group intention and awareness

What are the minimal levels of awareness and group intention needed for the agents on both sub-team and team levels?

The robots - two cases are applicable

1. They act only individually; this is the most limited (and economical) case;
2. They perform a limited form of cooperation, for example, they work together to clean up areas faster (e.g., by suitably combining the application of cleaning solids and catalysts), or pitch in for other robots when these turn out to be unable to perform their share of labor.

Both cases will be considered separately while investigating group attitudes of different types of agents involved in achieving a maintenance goal to keep the region safe.

The level of intention

1. In case 1, the robots need a general intention E-INT_{G_i} about the goals.
2. In case 2, $\text{E-INT}^2_{G_i}$ will be enough to allow forming two-robot teams that are not competitive internally. (But see [14] for a counter-example showing that a two-level intention is not sufficient to preclude competition among two-agent coalitions). If agents are supposed to be strictly cooperative, a two-level definition is also sufficient for larger teams: all agents intend to achieve the goal in cooperation with the others included in their team.

Although robots sometimes individually compete for resources, in our field of application where fast real-time team reaction to dangers is needed, we opt for strictly cooperative robots that use fixed protocols to load up on resources. The cleanup robots do not communicate with robots from other teams, and therefore do not need to have any beliefs, intentions and commitments regarding them.

The level of belief

1. In case 1, to act individually each robot i needs an individual belief about every group intention $(\mathrm{BEL}(i, \text{E-INT}_G(\varphi)))$. This way, a general belief $\text{E-BEL}_G(\text{E-INT}_G(\varphi))$ is in place and it suffices. Moreover, each robot in G_i should believe that the distribution of labor by means of bilateral commitments is done properly $(\text{E-BEL}_{G_i}(\bigwedge_{\alpha \in P} \bigvee_{n,m \in G_i} \mathrm{COMM}(n, m, \alpha)))$. This allows deliberation on actions of other robots from the same team. It may also prevent robots from doing all the work by themselves.
2. In case 2, $\text{E-BEL}^2_{G_i}$ will be enough to allow deliberation about other robots' intentions and beliefs (especially $\text{E-BEL}^2_{G_i}(\text{E-INT}^2_{G_i}(\varphi)))$. To see this, one may consider a pair of robots. With $\text{E-BEL}^2_{G_i}$, both robots have the same intention $(\text{E-INT}_{G_i}(\varphi))$, believe they have the same intention (the first-order belief $\text{E-BEL}_{G_i}(\text{E-INT}_{G_i}(\varphi)))$, and believe that the other believes this (the second-order belief $\text{E-BEL}_{G_i}(\text{E-BEL}_{G_i}(\text{E-INT}_{G_i}(\varphi)))$. Therefore, the robots can reason about the beliefs and intentions of their partner.

In both cases, it is assumed that the robots are incapable of forming coalitions of cardinality ≥ 2. This allows us to build a sufficiently strong collective intention based on a low degree of nesting of general intentions [13]. In case 2, the robots will also need to be aware of plan projections of their neighbours, in order to be able to notice when they can help.

The $\boldsymbol{UAVs}$ The $UAVs$ must sometimes work with one another. This requires at least E-BEL^2_G of other $UAVs$' intentions.

The level of intention - sub-team level Within each sub-team G_i consisting of an UAV and robots, UAV_i must make sure that all agents are motivated to do their tasks. Therefore:

– in case 1, $\mathrm{INT}(UAV_i, \text{E-INT}_{G_i}(\varphi))$ is required with respect to the sub-team intention $\text{E-INT}_{G_i}(\varphi)$,
– in case 2, $\mathrm{INT}(UAV_i, \text{E-INT}^2_{G_i}(\varphi))$ is required with respect to the sub-team intention $\text{E-INT}^2_{G_i}(\varphi)$. The UAV_i's intention is that all the robots in its team not only do their work, but also have the intention of helping each other in two-robot teams.

The level of belief - sub-team level Within each sub-team G_i consisting of UAV_i and robots $rob_{i_1}, \ldots, rob_{i_{n_i}}$, the UAV_i has the highest level of awareness, and acts as a coordinator. In order to facilitate this (make plans and reason correctly), it will require one level of belief more than its agents:

– in case 1, $\mathrm{BEL}(UAV_i, \text{E-BEL}_{G_i}(\text{E-INT}_{G_i}(\varphi)))$ is required with respect to the sub-team intention $\text{E-INT}_{G_i}(\varphi)$ as well as:
$\mathrm{BEL}(UAV_i, \text{E-BEL}_{G_i}(\bigwedge_{\alpha \in Cleanup} \bigvee_{i,j \in G_i} \mathrm{COMM}(i, j, \alpha))),$

– in case 2 $\mathrm{BEL}(UAV_i, \text{E-BEL}^2_{G_i}(\text{E-INT}^2_{G_i}(\varphi)))$ is required with respect to the sub-team intention $\text{E-INT}^2_{G_i}(\varphi)$ as well as:
$\mathrm{BEL}(UAV_i, \text{E-BEL}^2_{G_i}(\bigwedge_{\alpha \in Cleanup} \bigvee_{i,j \in G_i} \mathrm{COMM}(i, j, \alpha)))$.

The level of intention - team level There are situations when two $UAVs$ create coalitions of cardinality ≤ 2. In order not to be competitive internally, all $UAVs$ must have at least $\text{E-INT}^2_G(\varphi)$ with respect to every goal φ.

The level of belief - team level Cooperation between $UAVs$ requires at least E-BEL^2_G of other $UAVs$' intentions.

The coordinator The role of coordinator is to manage the team as a whole (see Figure 2), including all sub-teams and the pilot. Therefore he needs to know not only the global plan but also all the subplans.

The level of intention Similarly as in the relation of a UAV viz-a-viz its robots, the coordinator has one level of intention more than the $UAVs$ it manages, therefore $\mathrm{INT}(coordinator, \text{E-INT}^2_G(\varphi))$ is required.

The level of belief One extra level of belief allows the coordinator introspection and reasoning about the joint effort of all $UAVs$. Therefore, since teams are co-operative in a limited way, $\mathrm{BEL}(coordinator, \text{E-BEL}^2_G(\text{E-INT}^2_G(\varphi)))$ is required with respect to every group intention $\text{E-INT}^2_G(\varphi)$. Again, an analogical level of awareness is required with regard to distribution of bilateral commitments:
$\mathrm{BEL}(coordinator, \text{E-BEL}^2_G(\bigwedge_{\alpha \in Cleanup} \bigvee_{i,j \in G} \mathrm{COMM}(i, j, \alpha)))$.

Commands from the coordinator overrule temporary contracts between sub-teams. The coordinator does not only know the plan, but also keeps track of all relevant environmental conditions. It is assumed that even in the safe situation, the robots, the $UAVs$ and the pilot are prepared to take action at any moment.

The pilot The pilot's own awareness about the team level is similar to the $UAVs$', except that he usually does not need precise awareness about the robots' actions. Additionally, as a human team member he has a special position and we do not want to impose on him a straight-jacket of awareness from others. The only exception is that the $coordinator$ is aware of the pilot's share of labor and his commitments and if applicable, some $UAVs$ are aware of actions the pilot commits to perform for them.

6.5. Complexity of the language without collective attitudes

It seems that in the environmental case study, the language used is richer than propositional modal logic due to the use of continuous ranges, functions, and a theoretically unlimited number of agents and teams. Fortunately, most of the relevant part can be reduced to a fixed finite number of propositional atoms (that may be combined and be the subject of attitudes), based on finitely many predicates and constants, as follows:

- a fixed number of relevant environmental states;
- a fixed number of pre-named locations;
- a fixed finite number of agents and teams;
- a fixed finite number of other objects (liquids, solids, catalyst, helicopter);
- a fixed number of relevant thresholds $n_1, n_2, n_3, \epsilon_1, \epsilon_2$.

The only possible source of unbounded complexity is the use of continuous intervals and real-valued functions $f_1, f_2, f_3, \mathit{fit}$ appearing in Section 2. Recall that the architecture proposed in Section 1.2 allows us to query external entities. These concern data stored in databases and sensed from the environment, which are represented in the lower layer of the system. For example, the functions f_1, f_2, f_3 and *fit* are part of this lower layer. Therefore, even though the underlying structures are represented by first-order formulas, one extracts only propositional information from them to use in the upper layer of propositional TEAMLOG reasoning. In fact, one can obtain answers *true* or *false* about queries such as

$$f_3(p(A), t(A), c_1(A), c_2(A)) \in (v_3, n_3]?$$

from the lower layer, without needing to resort to a first-order language in the upper layer of the system.

7. Discussion of related approaches

One of the most influential theories of teamwork is the one of Wooldridge and Jennings [42]. The actual formal frameworks of their papers is quite different from ours. Wooldridge and Jennings define joint commitment towards φ in a more dynamic way than collective intentions defined in TEAMLOG: according to [42], initially the agents do not believe φ, and subsequently have φ as a goal until the termination condition is satisfied, including (as conventions) conditions on the agents to turn their eventual beliefs that termination is warranted into common beliefs. Subsequently, they define having a joint intention to do α as "having a joint commitment that α will happen next, and then α happens next". In contrast, agreeing with [5], we view collective commitments as stronger than collective intentions, and base the collective commitment on a specific social plan meant to realize the collective intention.

The emphasis on establishing appropriate collective attitudes for teamwork is shared with the SharedPlans approach of Grosz and Kraus [28, 27]. Nevertheless, the intentional component in their definition of collective plans is weaker than our collective intention: Grosz and Kraus' agents involved in a collective plan have individual intentions towards the overall goal and a common belief about these intentions; intentions with respect to the other agents play a part only at the level of individual sub-actions of the collective plan. We stress, however, that team members' intentions about their colleagues' motivation to achieve the overall goal play an important role in keeping the team on track even if their plan has to be changed radically due to a changing environment.

Similarly to [28, 27], Rao, Georgeff and Sonenberg [36] use a much weaker definition of joint intention than our collective intention: theirs is only one-level, being defined as "everyone has the individual intention, and there is a common belief about this". Thus, their definition does not preclude cases of coercion and competition. We have shown in the case study that, even though the full infinitary flavor of collective intention is not always needed, even among truly cooperative agents effective teamwork in disaster response demands at least a second level of general intentions (see section 6.4).

Another theory of teamwork, by Levesque, Cohen and Nunes [31], does incorporate higher-level intentions in its joint intention. Also, common belief is an integral part of a group's intention in both [31, 36], which may be problematic because creating common beliefs is costly in terms of communication and often impossible if the communication medium is untrustworthy [17, 4, 2]. Problematic communication was one of the reasons for restricting to minimal levels of belief in section 6.4. Additionally, in contrast to TEAMLOG, Levesque and colleagues assume a homogeneous group and the absence of social structure. Accounting for a group's social structure is crucial in a theory of teamwork, and in fact it is employed in our investigation of collective commitments in the case study.

Some approaches to collective commitments introduce other aspects, not treated here. For example, Aldewereld et al. add constraints about goal adoption and achievement to their definitions of joint motivational attitudes [1]. We have chosen to incorporate solely vital aspects of the defined attitudes, leaving room for any case-specific extensions. If needed, these extensions may be added by extra axioms. Note that in contrast to approaches such as [42, 31], our collective commitment is not iron-clad: it may vary in order to adapt to changing circumstances, in such a way that the collective intention on which it is based can still be reached.

In the logical MAS literature some phenomena such as the dynamics of attitude revision during reconfiguration have received scant attention (but see [38, 39]). Our notion of collective commitment ensures efficiency of reconfiguration in two ways. Unlike in [42], our approach to group commitments is formalized in a non-recursive way. This allows for a straightforward revision. Next, because only social commitments to individual actions appear, it often suffices to revise just some of them. That way it is possible to avoid involving the whole team in replanning. Thus, teamwork axioms may serve a system designer as a high-level specification at design-time. During run-time, formal verification methods may be applied to check the correctness of the system behavior.

In conclusion, our logical theory TEAMLOG has been developed against the background of Bratman's four criteria for shared cooperative activity [3]:
- mutual responsiveness;
- commitment to the joint activity;
- commitment to mutual support;
- formation of subplans that mesh with one another.

Kraus and colleagues [25] apply an illuminating comparison in the light of Bratman's criteria to six approaches to teamwork, including an early version of TEAMLOG [14] and some of those theories mentioned above [42, 28, 27, 31, 36] as well as [26]. They base their analysis on a very simple example of co-operation, where two agents move an object together. It would be interesting to compare all six theories on a complex time-critical example such as the present disaster management case study, but this would require a book-length study and is out of the scope of this paper.

8. Conclusion

In the case study we have shown how to implement teamwork within a strictly cooperative, but still heterogenous group of agents in TEAMLOG. The heterogeneity is taken seriously here, as advocated in [24]. Natural differences in agents' shares, opportunities and capabilities when acting together, have been reflected in different levels of agents' awareness about various aspects of their behaviour. The study dealt especially with cooperation and coordination. Having very generic definitions of common motivational and informational attitudes in TEAMLOG, it is challenging to choose the proper level of their complexity. We have shown that this is possible, by illustrating how to tailor complex definitions of intentions and commitments to a specific environment. Our focus was on building beliefs, intentions and, finally, commitments of all agents involved in teamwork on an adequate, but still minimal level. Therefore, even though not all aspects of teamwork have been shown, a bridge between theory and practice of teamwork has been constructed for this exemplary application, under the assumptions of fully cooperative agents and a hierarchical organization. It would be interesting to investigate the influence of relaxing both assumptions as well as to incorporate the dynamics of team dialogue and reconfiguration in the case study, according to the dynamic theory TEAMLOG^{dyn} [20, Chapters 5,6,8].

Future work will be to embed TEAMLOG into a form of approximate reasoning suitable for modeling perception. *Similarity-based approximate reasoning*, with its intuitive semantics compatible with that of TEAMLOG, is a promising candidate [9, 12, 10, 11].

References

1. Aldewereld, H., van der Hoek, W., Meyer, J.: Rational teams: Logical aspects of multi-agent systems. Fundamenta Informaticae 63 (2004)
2. van Baars, E., Verbrugge, R.: A communication algorithm for teamwork in multi-agent environments. Journal of Applied Non-Classical Logics 19, 431–461 (2009)
3. Bratman, M.: Shared cooperative activity. The Philosophical Review 101, 327–341 (1992)
4. Brzezinski, J., Dunin-Kęplicz, P., Dunin-Kęplicz, B.: Collectively cognitive agents in cooperative teams. In: Gleizes, M.P., Omicini, A., Zambonelli, F. (eds.) Engineering Societies in the Agents World V, (ESAW 2004): Revised Selected and Invited Papers. LNCS, vol. 3451, pp. 191–208. Springer, Berlin (2005)

5. Castelfranchi, C.: Commitments: From individual intentions to groups and organizations. In: Lesser, V. (ed.) Proc. First Int. Conference on Multi-Agent Systems. pp. 41–48. AAAI-Press and MIT Press, San Francisco (1995)
6. Cuny, F.: Disasters and Development. Oxford University Press, Oxford (1983)
7. Doherty, P., Granlund, G., Kuchcinski, K., Nordberg, K., Sandewall, E., Skarman, E., Wiklund, J.: The WITAS unmanned aerial vehicle project. In: Proc. of the 14th European Conference on Artificial Intelligence. pp. 747–755 (2000)
8. Doherty, P., Łukaszewicz, W., Skowron, A., Szałas, A.: Knowledge Representation Techniques. A Rough Set Approach, Studies in Fuzziness and Soft Computing, vol. 202. Springer Verlag (2006)
9. Doherty, P., Dunin-Kęplicz, B., Szałas, A.: Dynamics of approximate information fusion. In: Kryszkiewicz, M., Peters, J.F., Rybinski, H., Skowron, A. (eds.) RSEISP. Lecture Notes in Computer Science, vol. 4585, pp. 668–677. Springer (2007)
10. Dunin-Kęplicz, B., Nguyen, L., Szałas, A.: Fusing approximate knowledge from distributed sources. In: Papadopoulos, G., Badica, C. (eds.) Proceedings of the Third International Symposium on Intelligent Distributed Computing. Studies in Computational Intelligence, vol. 237, pp. 75–86 (2009)
11. Dunin-Kęplicz, B., Szałas, A.: Agents in approximate environments. In: van Eijck, J., Verbrugge, R. (eds.) Games, Actions and Social Software. College Publications, London (2010), to appear
12. Dunin-Kęplicz, B., Szałas, A.: Towards approximate BGI systems. In: H.-D.Burkhard, Lindemann, G., Verbrugge, R., Z.Varga, L. (eds.) CEEMAS. Lecture Notes in Computer Science, vol. 4696, pp. 277–287. Springer (2007)
13. Dunin-Kęplicz, B., Verbrugge, R.: Collective commitments. In: Tokoro [40], pp. 56–63
14. Dunin-Kęplicz, B., Verbrugge, R.: Collective intentions. Fundamenta Informaticae 51(3), 271–295 (2002)
15. Dunin-Kęplicz, B., Verbrugge, R.: Evolution of collective commitments during teamwork. Fundamenta Informaticae 56, 329–371 (2003)
16. Dunin-Kęplicz, B., Verbrugge, R.: A tuning machine for cooperative problem solving. Fundamenta Informaticae 63, 283–307 (2004)
17. Dunin-Kęplicz, B., Verbrugge, R.: Creating common beliefs in rescue situations. In: Dunin-Kęplicz, B., Jankowski, A., Skowron, A., Szczuka, M. (eds.) Proc. of Monitoring, Security and Rescue Techniques in Multiagent Systems (MSRAS). pp. 69–84. Advances in Soft Computing, Springer, Berlin (2005)
18. Dunin-Kęplicz, B., Verbrugge, R.: Awareness as a vital ingredient of teamwork. In: Stone, P., Weiss, G. (eds.) Proc. of the Fifth Int. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'06). pp. 1017–1024. IEEE/ACM Press, New York (NY) (2006)
19. Dunin-Keplicz, B., Verbrugge, R.: Dynamics of collective attitudes during teamwork. In: ESAW. pp. 107–122 (2003)
20. Dunin-Keplicz, B., Verbrugge, R.: Teamwork in Multi-Agent Systems: A Formal Approach. John Wiley & Sons, Ltd. (2010)
21. Dziubiński, M.: Complexity of the logic for multiagent systems with restricted modal context. In: Dunin-Kęplicz, B., Verbrugge, R. (eds.) Proc. of the Third Int. Workshop on Formal Approaches to Multi-Agent Systems, FAMAS'007. pp. 1–18. Durham University (2007)
22. Dziubiński, M., Verbrugge, R., Dunin-Kęplicz, B.: Complexity issues in multiagent logics. Fundamenta Informaticae 75(1-4), 239–262 (2007)
23. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning about Knowledge. MIT Press, Cambridge, MA (1995)

24. Gold, N. (ed.): Teamwork. Palgrave McMillan, Basingstoke and New York (2005)
25. Grant, J., Kraus, S., Perlis, D.: Formal approaches to teamwork. In: Artemov, S., others (eds.) We Will Show Them: Essays in Honour of Dov Gabbay, vol. 1, pp. 39–68. College Publications, London (2005)
26. Grant, J., Kraus, S., Perlis, D.: A logic-based model of intention formation and action for multi-agent subcontracting. Artificial Intelligence 163(2), 163–201 (2005)
27. Grosz, B., Kraus, S.: The evolution of SharedPlans. In: Rao, A., Wooldridge, M. (eds.) Foundations of Rational Agency, pp. 227–262. Kluwer, Dordrecht (1999)
28. Grosz, B., Kraus, S.: Collaborative plans for complex group action. Artificial Intelligence 86(2), 269–357 (1996)
29. Halpern, J., Moses, Y.: Knowledge and common knowledge in a distributed environment. Journal of the ACM 37, 549–587 (1990)
30. Kleiner, A., Prediger, J., Nebel, B.: RFID technology-based exploration and SLAM for search and rescue. In: Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS 2006). pp. 4054–4059. Bejing (2006)
31. Levesque, H., Cohen, P., Nunes, J.: On acting together. In: Proc. Eighth National Conference on AI (AAAI90). pp. 94–99. MIT Press, Cambridge (MA) (1990)
32. Meyer, J., van der Hoek, W.: Epistemic Logic for AI and Theoretical Computer Science. Cambridge University Press, Cambridge (1995)
33. Parikh, R., Krasucki, P.: Levels of knowledge in distributed computing. Sadhana: Proc. of the Indian Academy of Sciences 17, 167–191 (1992)
34. Pynadath, D.V., Tambe, M.: The communicative multiagent team decision problem: Analyzing teamwork theories and models. Journal of Artificial Intelligence Research 16, 389–423 (2002)
35. Rao, A., Georgeff, M.: Modeling rational agents within a BDI-architecture. In: Fikes, R., Sandewall, E. (eds.) Proc. of the Second Conference on Knowledge Representation and Reasoning. pp. 473–484. Morgan Kaufman (1991)
36. Rao, A., Georgeff, M., Sonenberg, E.: Social plans: A preliminary report. In: Werner, E., Demazeau, Y. (eds.) Decentralized A.I.-3. pp. 57–76. Elsevier, Amsterdam (1992)
37. Sycara, K., Lewis, M.: Integrating intelligent agents into human teams. In: Salas, E., Fiore, S. (eds.) Team Cognition: Understanding the Factors that Drive Process and Performance. pp. 203–232. American Psychological Association, Washington (DC) (2004)
38. Tambe, M.: Teamwork in real-world, dynamic environments. In: Tokoro [40], pp. 361–368
39. Tambe, M.: Towards flexible teamwork. Journal of Artificial Intelligence Research 7, 83–124 (1997)
40. Tokoro, M. (ed.): Proc. Second Int. Conference on Multi-Agent Systems. AAAI-Press, Menlo Park (CA) (1996)
41. Wisner, B., Blaikie, P., Cannon, T., Davis, I.: At Risk - Natural Hazards, People's Vulnerability and Disasters. Routledge, Wiltshire (2004)
42. Wooldridge, M., Jennings, N.: Cooperative problem solving. Journal of Logic and Computation 9, 563–592 (1999)

Barbara Dunin-Kęplicz s a Professor of computer science at the Institute of Informatics of Warsaw University and at the Institute of Computer Science of the Polish Academy of Sciences. She obtained her PhD in 1990 on computational linguistics from the Jagiellonian University, and in 2004 she was awarded her habilitation on formal methods in multi-agent systems from the Polish Academy of Sciences.

She is a recognized expert in multi-agent systems. She was one of the pioneers of modeling BDI systems, recently introducing approximate reasoning to the agent-based approach.

Rineke Verbrugge is a Professor of logic and cognition at the Institute of Artificial Intelligence of the University of Groningen. She obtained her PhD in 1993 on the logical foundations of arithmetic from the University of Amsterdam, but shortly thereafter moved to the research area of multi-agent systems.

She is a recognized expert in multi-agent systems and one of the leading bridge-builders between logic and cognitive science.

Michał Ślizak is a PhD student at the Institute of Informatics of Warsaw University, studying the field of multi-agent systems and approximate reasoning under supervision of Barbara Dunin-Keplicz.

Received: February 9, 2010; Accepted: May 3, 2010.