Collective Obligations and Agents: Who Gets the Blame?

D. Grossi[1], F. Dignum[1], L. Royakkers[2], and J.-J. Ch. Meyer[1]

[1]Utrecht University,
The Netherlands
{davide,dignum,jj}@cs.uu.nl

[2]Eindhoven University of Technology,
The Netherlands
L.M.M.Royakkers@tm.tue.nl

1   Introduction

In multi-agent systems, the cooperation between agents is an important issue. For that purpose several theories have been developed about joint goals, plans and intentions. All these joint attitudes try to capture some of the team elements of agents that work together. In this paper we follow up on that work and will look at collective obligations. What are the consequences for an agent if the group in which an agent performs its tasks gets an obligation to fulfill a certain goal? E.g. a program committee of DEON'04 may have the collective obligation to review all submitted papers before a certain time. We are interested to explore how this collective obligation translates into individual obligations for the program committee members, e.g. review two or three papers, and the extra obligations for the program chair to divide the papers and monitor the process and make final decisions.

In [11], the collective obligation is formalized in a framework of deontic logic, which gives the opportunity to express which agent (or which group of agents) has the responsibility to bring about a certain situation (to express group liability, e.g. liability for a trading partnership) and to express the relation between the agents of a group. However, in the theory developed in [11] we cannot express the individual responsibility that follows from the task to achieve the fulfillment of the collective obligation. A consequence is that we cannot indicate which individual is responsible for a violation of a collective obligation.

We believe that the main differences between individual and collective obligations can be explained through the distribution of *responsibility* and *knowledge*.

If an individual has an obligation he has to perform all tasks for the fulfillment of the obligation (including planning the tasks) himself. Therefore whenever the obligation is not fulfilled the cause is that the individual did not perform a task that should be done to fulfill the obligation. It follows that, whatever the actual task was that was not performed, the individual is responsible. This differs fundamentally from the situation where a group has to fulfill the obligation. In this case the tasks are distributed over the group. Each individual can independently

(autonomously) decide to perform his task or not. The responsibility therefore also is distributed over the members of the group. If the group does not fulfill the obligation the individual that did not perform his task will be held responsible! So, it becomes important to check exactly which are the obligations for each member of the group that follow from the collective obligation of that group.

The second aspect that is very different between collective and individual obligation is the distribution of knowledge. An individual can decide for himself whether to fulfill an obligation or not, based on its utility, beliefs, goals etc. The individual also knows the complete plan, whether he is capable to perform the actions in the plan, when he has performed actions or decided not to perform them. All this information is readily available and can be reasoned about. However, when a collective obligation is divided over the individuals of that collective, they might not know the whole plan, typically do not have information about actions that are performed, etc.

We make the important assumption that an individual can only be responsible for violating an obligation if he knows (or could have known) that he has the obligation. If a group has a collective obligation, but a member of the group does not know what are the consequences of that obligation for himself, he cannot be held responsible for violating that part of the obligation (unless he knows he has the obligation to find out what he has to do, but never bothered to do it). This assumption has many consequences for the desired dissemination of knowledge through the group about the plan, plan allocation and current execution of the plan, etc. In the ideal case all members of the group have complete knowledge about all these aspects, such that no member can avoid his responsibility based on a lack of knowledge. It leads to the introduction of explicit coordination actions in our model to incorporate this aspect of the collective obligation. We will discuss these in detail in Section 2.

Coordination actions are actually only one type of meta actions that should be considered. Besides the plan to achieve the content of the obligation the group should create that plan, allocate agents to parts of the plan, create a plan for what to do when the original plan fails, etc. These meta actions should also be coordinated again creating in the end an infinite regression of meta actions. In this paper we will not take all these layers into account, but will limit us to the coordination actions that are necessary during the execution of a plan to fulfill the obligation.

In the next section we will discuss the notions of collective agency, plans, task allocations and coordination of tasks. This will indicate which concepts are needed in the formal framework to describe collective obligations and their consequences. The formal framework is described in Section 3. In Section 4, we show how the formal framework can be used to model an example and how some consequences of collective obligations can be derived depending on characteristics of the obligated task, structure of the group and their knowledge. In Section 5 we draw some conclusions and give directions for future research.

2 Conceptualizing Collective Agency: Plans, Allocation, Coordination

In this section we discuss two basic issues that underly the analysis of the consequences of a collective obligation for the individuals of the group. First we discuss the notion of a (distributed) plan to achieve the collective obligation. After that we discuss some of the coordination issues around these distributed plans. During the discussion we will make use of the notation of the formal framework that is fully described in Section 3. The intuitions and properties described here do not depend on this formalism though!

2.1 Concepts of plan and task allocation

To fulfill a collective obligation $O(X : \gamma)$, the group X has to perform a complex action γ^1. The concrete manner to deal with such a complex action is planning. The group has to decompose the complex action into a number of individual subactions. For example, if the program committee is obliged to notify the authors of the submitted papers of acceptance before a certain deadline, this obligation can only be fulfilled if the work that has to be done, is shared out in several tasks over the members of the program committee. Therefore, the group needs a plan: a concrete manner to achieve the task of the group.

We can define a plan to perform the complex action γ as a decomposition of the complex action γ by a sequence of (possible simultaneous) individual actions:

$$Plan(\gamma) = \langle \alpha_1 \bullet \alpha_2 \bullet \ldots \bullet \alpha_n \rangle \text{ such that } \gamma = \alpha_1 \bullet \alpha_2 \bullet \ldots \bullet \alpha_n,$$

where $\bullet$ stands for the simultaneous operator '&' or the sequential operator ';'. The action $\alpha_1 \& \alpha_2$ stands for the simultaneous performance of α_1 and α_2, and action $\alpha_1; \alpha_2$ stands for the sequential composition of α_1 and α_2.

We need the simultaneous operator, since some actions have to be performed at the same time. The sequential operator is needed for the fact that some actions might depend on other ones: a certain action can only be performed if an other action is done. So, the plan must at least determine the *order* of subactions. For example, the notification of acceptance of a certain paper can only be done if it is reviewed by the delegated members of the program committee. The responsibility of the performance of an action α by an agent depends not only on the individual who is committed to perform the action α, but also on agents who have to perform actions which are necessary to perform action α.

The complex action γ does not contain a non-deterministic choice, because we only use totally ordered plans [14]. However, there can be several totally ordered plans to execute a certain action γ, and it makes therefore sense to talk about a choice of plans. In this paper we restrict ourselves to *acceptable* plans, which are plans containing individual actions that can be performed by an agent

---

[1] Such a complex action γ may be seen as equivalent to an action of the type $achieve(\tau)$, where τ is a state of affairs.

in the group. In other words, for every individual action α there is an agent of the group which is *capable* to perform α. Note that there are better and worse plans. E.g. a plan that consists of individual actions that only can be performed by one agent seems worse than a plan that consists of individual actions that can be performed by several agents. Since there are several possible ways to divide the task, the group has to decide which plan should be followed.

Besides task division (the decomposition of a complex actions γ into individual sub-actions $\alpha_1, \ldots, \alpha_n$), task allocation is needed, which indicates which (capable) agent of the group has to perform which sub-action of the complex action. We use the following definition for (partial) task allocation:

Definition 1. (Partial task allocation)
A partial task allocation for a task γ within a group of agents X is defined as follows:

$$< I_1 : \alpha_1 \ \bullet \ I_2 : \alpha_2 \ \bullet \ \ldots \ \bullet \ I_n : \alpha_n > \ \ such \ that$$

$$\gamma = \alpha_1 \bullet \alpha_2 \bullet \ \ldots \ \bullet \alpha_n, \ and \ I_j \in \mathcal{P}^+(X) \ \ for \ \ j = 1, 2, \ldots, n.$$

We refer to the set of all partial task allocations of γ within X as $PPlan(X : \gamma)$

A partial task allocation is thus a composition of constructs of the type "group I performs the (possibly complex)action α" $(I : \alpha)$. These constructs are called *(collective) events* ([11]).

Definition 2. (Complete task allocation)
CP is a complete task allocation for a task γ to a group X, iff $CP \in PPlan(X : \gamma)$ and for all I_j occurring in CP holds that $I_j = \{a_j\}$[2].

E.g. let $CP \in PPlan(\{a_1, a_2, a_3\}, \gamma)$ and $CP = \langle a_1 : \alpha_1 \ ; \ a_2 : \alpha_2 \ \& \ a_3 : \alpha_3 \rangle$. In CP agent a_2 is responsible for the performance of action α_2, but according to the plan he can only perform his action after agent a_1 has done action α_1. The same holds for agent a_3.

The collective obligation $O(X : \gamma)$ can be translated by the group X in internal obligations of the group given a plan. E.g. given that we have $O(\{a_1, a_2, a_3\}, \gamma)$ and plan CP above. Then we would at least have that:

$$O(a_1 : \alpha_1) \wedge [a_1 : \alpha_1](O(a_2 : \alpha_2) \wedge O(a_3 : \alpha_3))$$

Which states that first a_1 is obliged to perform α_1 and if a_1 has performed α_1 ($[a_1 : \alpha_1]$) then a_2 and a_3 are obliged to perform α_2 and α_3 respectively. In general, we have that given a complex collective obligation, a disjunction of all internal obligations, which are determined by the possible acceptable plans, can be derived. The fulfillment of one of these internal obligations leads to the fulfillment of the collective obligation. Violation of the internal obligation leads, typically, to a decision to make another plan establishing an alternative internal obligation, or to the violation of the collective obligation itself.

---

[2] In what follows, in order to keep formal expressions more readable, we will often refer to singletons omitting the {} standard notation.

The above indicates how a plan can be used to distribute a collective obligation over the members of a group. However, it also shows that the obligations of at least some members (in the example above i_2 and i_3) depend on the performance of an action of another member of the group (in the example above i_1). This observation leads us to the issue of coordination between the actions in a plan.

2.2 The problem of coordination

The view on coordination, which is assumed here, can be perfectly summarized by the following quotation from [4]:

> "coordination is the process of managing interdependencies between activities"'.

Plans and task allocations, as they have been defined in Section 2.1, are essentially sequences of, respectively, simpler actions and events. This sequential structure is precisely what should be managed by means of coordination. Let us reconsider our example about the group organizing the DEON'04 workshop: for each submitted paper a reviewing process takes place, which ends, in some cases, with a notification of acceptance. The first question coming naturally about is: when should the agent responsible for notifying acceptances start his activity? I.e. how does he *know* that the condition for his obligation to send the notifications is true? In the introduction we mentioned that we only consider an agent responsible for violating an obligation when he knows he has the obligation. We assume that if an agent knows the plan to fulfill the obligation and knows that the actions he depends on are performed he will also know that he has an obligation to perform his own part. Using the example and the notation used before plus B_i to denote the beliefs of agent a_i and $DONE(a : \alpha)$ to indicate that agent a has performed α we get:

$$B_2([a_1 : \alpha_1](O(a_2 : \alpha_2)) \wedge B_2(DONE(a_1 : \alpha_1)) \rightarrow B_2(O(a_2 : \alpha_2)) \qquad (1)$$

This suggests that a_2 should somehow come to believe that a_1 performed its task. We do assume that an agent believes it performed a task whenever it did so (internal monitoring or conscious behavior) but this leads only to:

$$[a_i : \alpha]DONE(a_i : \alpha) \wedge B_{a_i}DONE(a_i : \alpha)$$

and not to:

$$[a_i : \alpha]DONE(a_i : \alpha) \wedge B_{a_j}DONE(a_i : \alpha)$$

for any $j \neq i$.

The latter can only be achieved through the introduction of explicit coordination actions. This constraint determines the necessity for agents to be informed *at least* about what the previous agent did. And such a necessity impinges on the task decomposition itself forcing it to take into consideration how to provide agents with the necessary information to act according to the established plan.

A task allocation $a_1 : \alpha$; $a_2 : \beta$ for the complex action γ for the group X, once taken into account this type of coordination problem concerning informational issues, becomes something of this kind:

$$a_1 : \alpha \; ; \; a_i : \text{coordinate}(\text{a}_1, \alpha, \text{Y}) \; ; \; a_2 : \beta$$

where $a_i \in X$ and $Y \subseteq X$ with $Y \neq \emptyset$. The *coordinate* action is here understood as an action after each execution of which a belief holds in each agent belonging to Y which concerns an activity executed by the previous agent. Notice that Y should always be such that it contains at least the agent responsible for the task that follows within the task allocation sequence: so if a_{j-1} is the preceding agent, Y should be such that $a_j \in Y$. Notice moreover how the coordination action assumes different intuitive meanings depending on the agent appointed to its performance. In the example above we have that possibly $a_i = a_1$ or $a_i = a_2$. In the first case the *coordinate* action becomes a kind of *informative*, the agent itself being the one providing information about its performance:

$$[a_{j-1} : \alpha][a_{j-1} : \text{coordinate}(a_{j-1}, \alpha, Y)] \bigwedge_{a_k \in Y} B_{a_k} DONE(a_{j-1} : \alpha)$$

and $\exists a_k \in Y$ s.t. $a_k = a_j$. In the second case the *coordinate* action turns out to correspond to a sort of *checking*, since it is the agent himself which acquires the necessary information before acting:

$$[a_{j-1} : \alpha][a_j : \text{coordinate}(a_{j-1}, \alpha, Y)] \bigwedge_{a_k \in Y} B_{a_k} DONE(a_{j-1} : \alpha)$$

and $\exists a_k \in Y$ s.t. $a_k = a_j$. Notice, in the end, that even a "third party" might be responsible for this coordination task.

Due to the introspection principle for agents concerning their actions the coordination action between two actions of the same agent becomes superfluous and takes the form of a so-called *skip* action:

$$a_i : \alpha \; ; \; a_i : \text{coordinate}(a_i, \alpha, \{a_i\}) \equiv a_i : \alpha \; ; \; a_i : skip.$$

The property expressed in formula (1) relates the believes of an agent about the plan and the directly preceding actions to his own obligations. One might also be interested in the fact that an agent that is obliged to perform an action on which your actions depends does not perform his action. In the above example, in which $Y = \{a_2, a_3\}$, we could have:

$$O(a_1 : \alpha_1) \wedge [\overline{a_1 : \alpha_1}][a_j : \text{coordinate}(a_1, \overline{\alpha_1}, \{a_2, a_3\})]$$
$$(B_{a_2}(\neg DONE(a_1 : \alpha_1)) \wedge B_{a_3}(\neg DONE(a_1 : \alpha_1)).$$

where $\overline{a_1 : \alpha_1}$ denotes the negation of an event. This coordination task takes care that both in the case when an action is performed as scheduled as well as when an action is not performed as scheduled the dependent agents get to know about it. Further coordination mechanisms are usually devised which remedy

the violation of an obligation by one of the agents ([15]). We will not get into this aspects in this paper yet.

In the next section we will describe the formal framework in which the discussions above can be expressed formally and which provides the right validities to indeed prove the properties that we would like to have in order to distribute the collective obligations over the individuals of a group.

3 A formal framework

Our aim is to further pursue a line of analysis, which has been already proposed in [12, 11], trying to provide a formal framework in which to account for the problems of collective obligation and collective agency in terms of the concepts introduced in the previous section. Some attempts to formally address these problems have been already proposed ([1–3]), but none of the frameworks presented so far possesses the necessary expressive power for dealing with concepts such as plan and task allocation. One of the main reasons is that they are basically grounded on *stit* or *bringing-it-about* action logics, which cannot cope with any specification of the internal structure of plans.

More technically, this framework should handle event expressions, that is expressions about the performance of some action by some agents, and their composition, doxastic expressions, deontic expressions concerning event expressions, predicates on events. The framework is obtained expanding the proposal contained in [11] in several directions: first of all adding transactions and negations on transactions, following the line described in [5]; then adding doxastic logic; and finally adding a type of dynamic assertions, by means of the $DONE$ operator, in order to express, in analogy with [6, 7], performances actually (and not just possibly) taking place in a backward direction.

3.1 Language

The alphabet consists of a set P of propositional symbols (p), the propositional constant V, the operator $DONE$, a set of agent identifiers I (groups of agents identifiers are denoted by $X, Y, \ldots$), a set A of atomic action symbols typically denoted by $\underline{a}$ (this set at least includes the coordination actions $coordinate(a_i, \alpha_i, Y)$ with $a_i \in I$), the doxastic operator B, and the dynamic operators $[\,]$ and $\langle\,\rangle$. The language $\mathcal{L}$ is based on three types of syntactic constructs that we are now going to define.

The set Act of action expressions (α) is defined through the following BNF:

$$\alpha ::= \underline{a} \mid skip \mid \overline{\alpha} \mid \alpha_1 + \alpha_2 \mid \alpha_1 \& \alpha_2 \mid \alpha_1 ; \alpha_2.$$

where $skip$ represents a "doing nothing" action, $\dot{-}$ stands for the negation operator, + stands for the indeterministic choice operator, & for the parallel performance operator and ; for the sequencing operator. The set Evt of event expressions (ξ) is defined through the following BNF:

$$\xi ::= X : \alpha \mid \overline{X : \alpha} \mid \xi_1 + \xi_2 \mid \xi_1 \& \xi_2 \mid \xi_1 ; \xi_2.$$

Notice that the same notation for actions and event operators (negation, $+, \&, ;$) is used. It is nevertheless obvious that they belong to different categories of operators. We chose, however, for keeping notation not too rich.

The set Ass of assertions (ϕ) is defined through the following BNF:

$$\phi ::= p \mid V \mid DONE(\xi) \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \phi_1 \wedge \phi_2 \mid \phi_1 \rightarrow \phi_2 \mid [\xi]\phi \mid B_i\phi.$$

3.2 Models

In order to give a semantics to the language introduced above we start defining the notion of model for $\mathcal{L}$.

Definition 3. (Models) *A model M is defined as follows:*

$$M = \left\langle \mathcal{P}^+(\mathbb{I}), \mathbb{A} \cup \mathrm{skip}, \mathbb{W}, [\![\ ]\!]_R, \{\mathbb{R}_i\}_{i\in\mathbb{I}}, \prec, \pi \right\rangle$$

where:

- $\mathcal{P}^+(\mathbb{I})$ *is the non-empty powerset of the finite set of actors $\mathbb{I}$, that means the possible groups of actors. We assume $\mathbb{I} = I$.*
- $\mathbb{A} \cup \mathrm{skip}$ *is the set of actions.*
- $\mathbb{W}$ *is the set of possible states.*
- $[\![\ ]\!]_R$ *is a function f s.t. $f : Evt \times \mathbb{W} \longrightarrow \mathcal{P}(\mathbb{W})$, to each event expression-world couple it associates the set of states to which the performance of that event in that world leads. It consists of a composition of the two functions $[\![\ ]\!]$ and R which will be introduced in Section 3.3.*
- $\{\mathbb{R}_i\}_{i\in\mathbb{I}}$ *is a family of serial symmetric and transitive accessibility relations which are indexed by actors indicating the believable worlds of agent a_i[3].*
- $\prec$ *is a partial ordering on $\mathbb{W}$ denoting the order in which worlds are reached through actual performances of events. This ordering is constrained as follows: if $w_1 \prec w_2$ and $\exists w_3$ s.t. $w_3\mathbb{R}_iw_1$ or $w_3\mathbb{R}_iw_2$ then $w_3\mathbb{R}_iw_2$ and $w_3\mathbb{R}_iw_1$[4]. From an intuitive point of view, this condition guarantees the whole path of actual performances through $\mathbb{W}$ to be doxastically accessible.*
- π *is a usual truth function f s.t. $f : Ass \times W \longrightarrow \{1, 0\}$.*

Like in [8, 5, 6] our semantics consists of two parts: first event expressions are interpreted as set theoretic constructs on $\mathbb{A}$ where events get a so-called *open* interpretation; successively event expressions are interpreted as state-transition functions determining the accessibility relation $[\![\ ]\!]_R$ on $\mathbb{W}$.

3.3 Synchronicity sets, steps, synchronicity traces, and worlds

The interpretation of events is based on the basic notion of *synchronicity set* (s-set).

---

[3] The doxastic part of the framework will be of interest in particular for the modelling of an example in Section 4.

[4] This condition is important for proving validity (10) in Proposition 1.

Definition 4. (s-set)
The set $\mathcal{S}$ of s-sets is defined as follows: $\mathcal{S} = \mathcal{P}^+(\mathbb{I}) \times \{\text{skip}\} \cup \mathcal{P}^+(\mathbb{I}) \times \mathcal{P}^+(\mathbb{A})$.

Synchronicity sets, that is elements of $\mathcal{S}$, are denoted by $S_1, S_2, \ldots$. Informally, a s-set is nothing but a set of parallel executions of events by a group of agents, and formalises the aforementioned open interpretation view on events. Based on the notion of s-set we define the notion of *step* [5].

Definition 5. (Step)
The set Step of steps is defined as follows:

$$Step = \{\times_{X \in \mathcal{P}^+(\mathbb{I})} \mathcal{S}_X \mid \forall X, Y \in \mathcal{P}^+(\mathbb{I}) : Y \subseteq X \Rightarrow act(S_Y) \subseteq act(S_X) \ \&$$
$$\forall X, Y \in \mathcal{P}^+(\mathbb{I}) : act(S_Y) = \text{skip} \Rightarrow act(S_{X \cup Y}) = act(S_X)\}$$

where act is a function that extracts the action component from a given s-set $(act(X : \{a_1, a_2\}) = \{a_1, a_2\})$.

Steps represent a sort of snapshot of the activity of each subgroup of $\mathbb{I}$ at a certain moment, depicting how all agents move one "step" ahead. Steps are therefore sets of s-sets of cardinality $2^n - 1$ where n is the number of agents in $\mathbb{I}$. They are constrained in such a way that whatever action is performed by a subgroup is also performed by a supergroup, and subgroups remaining inactive are treated as performing a skip action. Steps, that is elements of *Step*, are denoted by $s_1, s_2, \ldots$.

In order to provide a semantics for sequential expressions the concept of *synchronicity trace* (s-trace) is needed. Notice that this concept uses steps instead of s-sets like in [8, 5].

Definition 6. (s-trace) *The set $\mathcal{T}$ of s-traces is defined as follows:*

$$\mathcal{T} = \{\langle s_1, ..., s_n, ... \rangle \mid s_1, ..., s_n, ... \in \mathcal{S}\}.$$

The length of an s-trace t is denoted by $dur(t)$. We assume $dur(t)$ to be finite.

An event will be interpreted as a set of s-traces. The range for our interpretation of events is a set $\mathcal{E}$ such that $\mathcal{E} = \mathcal{P}(\mathcal{T})$. Elements of $\mathcal{E}$ (sets of s-traces) are denoted as $T_1, T_2, \ldots$ The length $dur(T)$ of a set T is defined as $max\{dur(t) \mid t \in T\}$.

We can now introduce the operations that constitute the semantic counterpart of our syntactic operators.

Definition 7. (Operations on events) *Let $T_1, T_2 \in \mathcal{T}$:*

$$T_1 \circ T_2 = \{t_1 \circ t_2 \mid t_1 \in T_1, t_2 \in T_2\}$$
$$T_1 \cap T_2 = \bigcup \{t_1 \cap t_2 \mid t_1 \in T_1, t_2 \in T_2\}$$
$$T_1 \uplus T_2 = T_1 \cup T_2 - (\bigcup\{t_1 \cap t_2 \mid t_1 \in T_1, t_2 \in T_2 \ \ and \ \ t_1 \neq t_2\})$$
$$\tilde{T} = \begin{cases} if \ T \neq \emptyset, \tilde{T} = \cap\{\tilde{s} \mid s \in T\} \\ if \ T = \emptyset, \tilde{T} = Step \end{cases}$$

---

[5] Notice that in [6] s-sets are called steps, and no notion of step as it will be defined in this work occurs there.

Where:

- $t_1 \circ t_2$ *is defined as follows: if* $t_1 = \langle s_1, ..., s_n \rangle$ *and* $t_2 = \langle s'_1, ..., s'_m \rangle$ *then,* $t_1 \circ t_2 = \langle s_1, ..., s_n, s'_1, ..., s'_m \rangle$.

- $t_1 \pitchfork t_2$ *is defined as follows:* $t_1 \pitchfork t_2 = \begin{cases} t_1 & \text{if } t_2 \in start(t_1) \\ t_2 & \text{if } t_1 \in start(t_2) \\ \emptyset & \text{otherwise} \end{cases}$

  The function start associates to a given s-trace all its starting possible s-traces.

- $\tilde{t}$ *is defined as follows:* $\tilde{t} = \bigcup_{1 \leq n \leq dur(t)} \langle s_1, ..., \tilde{s_n} \rangle$, *where* $\tilde{s} = Step - \{s\}^6$.

Intuitively, we want $\uplus$ to yield the property: $a \equiv a + a; b$ for event expressions. In order to establish this property we cannot just use a union of the sets of s-traces representing a and $a; b$ but have to do some "cleaning up" by subtracting superfluous parts.

The semantics of events are obtained by means of a function $[\![\ ]\!] : Evt \longrightarrow \mathcal{E}$ such that:

Definition 8. (Semantics of events)

$$[\![X : \underline{a}]\!] = \{s \mid S_X \in s, a \in act(S_X)\}$$
$$[\![\xi_1; \xi_2]\!] = [\![\xi_1]\!] \circ [\![\xi_2]\!]$$
$$[\![\xi_1 + \xi_2]\!] = [\![\xi_1]\!] \uplus [\![\xi_2]\!]$$
$$[\![\xi_1 \& \xi_2]\!] = [\![\xi_1]\!] \pitchfork [\![\xi_2]\!]$$
$$[\![\bar{\xi}]\!] = [\![\tilde{\xi}]\!]$$
$$[\![skip]\!] = \{\text{skip}\}.$$

The basic clause stipulates that the meaning of an atomic event consists of the set of steps where that action at least is performed by that specific group of agents.

On the basis of this evaluation for events, an evaluation of groups performing complex actions is obtained:

Definition 9. (Semantics of collective actions)

$$[\![X : \alpha_1; \alpha_2]\!] = [\![X : \alpha_1]\!] \circ [\![X : \alpha_2]\!]$$
$$[\![X : \alpha_1 + \alpha_2]\!] = [\![X : \alpha_1]\!] \uplus [\![X : \alpha_2]\!]$$
$$[\![X : \alpha_1 \& \alpha_2]\!] = [\![X : \alpha_1]\!] \pitchfork [\![X : \alpha_2]\!]$$
$$[\![X : \overline{\alpha'}]\!] = \overline{[\![X : \alpha']\!]}.$$

To connect this interpretation of events to a possible world semantics a function $R : \mathcal{E} \times \mathbb{W} \longrightarrow \mathbb{W}$ is defined, which couples events with state-transitions.

---

[6] Negation of sequences constitutes a delicate matter. For a deeper discussion of this issue we refer to [5].

Definition 10. (Function R)
$R(T, w_1) = \{w_2 \mid \exists t \in T \quad s.t. \quad w_2 = R(t, w_1)\}$ *where R on transitions is inductively defined as follows:*

$$R(s_1, w_1) = r(s_1, w_1)$$
$$R(t_1 \circ t_2, w_1) = R(t_2, R(t_1, w_1)).$$

and $r : \mathcal{S} \times \mathbb{W} \longrightarrow \mathbb{W}$, that is a function that, given a state, returns the following state reachable through a given synchronicity set, and such that $r(\{\mathrm{skip}\}, w) = w$.

3.4 Evaluating formulas

The meaning of formulas ϕ in a world w, given the structure M, is defined as usual. For space reasons we report here only clauses for dynamic operators, and the $DONE$ unary operator.

Definition 11. (Semantics of assertions) *In the following let $dur(\llbracket \xi_1 \rrbracket) = 1$,*

$M, w_1 \models [\xi]\phi \quad iff \quad \forall w_2 \in \llbracket \xi \rrbracket_R(w_1), M, w_2 \models \phi$

$M, w_1 \models \langle \xi \rangle \phi \quad iff \quad \exists w_2 \in \llbracket \xi \rrbracket_R(w_1), M, w_2 \models \phi$

$M, w_1 \models DONE(\xi_1) \quad iff \exists w_2 \in \mathbb{W}, \ w_2 \prec w_1 \Rightarrow w_1 \in \llbracket \xi_1 \rrbracket_R w_2;$

$M, w_1 \models DONE(\xi; \xi_1) \quad iff \quad \exists w_2 \in \mathbb{W}, \ w_2 \prec w_1 \Rightarrow M, w_1 \models DONE(\xi_1) \quad and$
$\qquad M, w_2 \models DONE(\xi).$

Informally, a sentence $[\xi]\phi$ ($\langle \xi \rangle \phi$) is true in w iff ϕ is true in every world (respectively in at least one world) accessible through a performance of ξ. Sentences $DONE(\xi)$ are evaluated as true in a world w iff that world may be reached from w_n performing ξ, and ξ is actually performed in w_n and all intermediate worlds w_i are reachable from w_n performing starting sequences of $\llbracket \xi \rrbracket$ of length $n - i$ along the $\prec$ ordering.

We do not provide a separate semantics for the coordination actions. Instead we give the following constraint on the possible models which indicates that the effect of a coordination action is a certain type of believe in the recipients of that action:

Definition 12. (Coordination action) *Let $a_j \in \mathbb{I}$ with $1 \leq j \leq n$ and $n = ||\mathbb{I}||$, then:*

$M, w_1 \models DONE(\{a_1\} : \alpha_1) \quad \Rightarrow \quad \forall w_2 \in \llbracket \{a_i\} : coordinate(a_1, \alpha_1, Y) \rrbracket_R(w_1) :$
$$M, w_2 \models \bigwedge_{a_j \in Y} B_j(DONE(a_1 : \alpha_1))$$

The deontic notions, which range over events, are defined according to the following reduction principles:

Definition 13.

$$F(\xi) \equiv [\xi]V;$$
$$O(\xi) \equiv [\overline{\xi}]V;$$
$$P(\xi) \equiv \neg[\xi]V.$$

For an extensive account of how these notions are related we refer to [11]. In what follows our attention is exclusively focused on expressions concerning obligations.

3.5 Some Relevant Validities

We will now focus on some validities that are relevant in order to model the collective obligations appropriately. The validities listed below are of three kinds: validities (2)-(9) show how obligations propagate within the group, that is how the sub/supergroup relation is connected with the enactment of a group obligation and in particular how the obligation on a plan distributes over the obligations on the single components of that plan; second, validity (10) has to do with the beliefs of each individual agent about the task allocation and related obligations addressed to him; third, validities (11)-(14) are of a more general kind, and express some very basic features of the framework.

Proposition 1. (Validities) *Let* $X, Y \in \mathcal{P}^+(\mathbb{I})$ *and* $\alpha_1, \alpha_2, \gamma_1, \gamma_2 \in Act$ *and* γ_1, γ_2 *do not contain any occurrence of the action negation symbol:*

$$\models O(X \cup Y : \overline{\gamma_1}) \rightarrow O(X : \overline{\gamma}) \tag{2}$$

$$\models O(X : \gamma) \rightarrow (X \cup Y : \gamma) \tag{3}$$

$$\models O(X : \overline{\gamma_1} + Y : \overline{\gamma_1}) \rightarrow O(X \cap Y : \overline{\gamma_1}), \quad with \ \ X \cap Y \neq \emptyset \tag{4}$$

$$\models O(X : \gamma_1 + Y : \gamma_1) \rightarrow (X \cup Y : \gamma) \tag{5}$$

$$\models O(X : \overline{\gamma_1} \ \& \ Y : \overline{\gamma_2}) \rightarrow (X \cap Y : \overline{\gamma_1} \& \overline{\gamma_2}), \quad with \ \ X \cap Y \neq \emptyset \tag{6}$$

$$\models O(X : \gamma_1 \ \& \ Y : \gamma_2) \rightarrow (X \cup Y : \gamma_1 \& \gamma_2) \tag{7}$$

$$\models O(X : \gamma_1 \ ; \ Y : \gamma_2) \rightarrow O(X \cup Y : \gamma_1; \gamma_2) \tag{8}$$

$$\models O(X : \alpha_1 \ \& \ Y : \alpha_2) \leftrightarrow O(X : \alpha_1) \wedge O(Y : \alpha_2) \tag{9}$$

$$\models B_i(O(\{j\} : \alpha_1 \ ; \ \{i\} : \alpha_2)) \rightarrow B_i(DONE(\{j\} : \alpha_1) \rightarrow O(\{i\} : \alpha_2)) \tag{10}$$

$$\models O(X : \alpha_1 \ ; \ Y : \alpha_2) \leftrightarrow O(X : \alpha_1) \wedge [X : \alpha_1]O(Y : \alpha_2) \tag{11}$$

$$\models O(X : \alpha_1) \vee O(Y : \alpha_2) \rightarrow O(X : \alpha_1 + Y : \alpha_2) \tag{12}$$

$$\models [X : \alpha_1]DONE(X : \alpha_1). \tag{13}$$

Proofs are omitted but can be easily obtained from the semantics presented. Nevertheless, some of these validities deserve some remarks. We start from (10), which shows that, given a (possibly limited) knowledge of the strategy of the group, that is, of a task allocation sequence, and given the knowledge of what just happened, an individual agent is aware of the obligations stemming from that task allocation and addressed to him. This validity is in some sense central

for both coordination and knowledge problems raised in the Introduction and in section 2.

Other interesting validities are (3) and (8), showing how each action of a subgroup is also an action of a supergroup (this manifests exactly the constraints contained in Definition 5). To get from the action of a supergroup to the ones of the subgroups is a more complicated matter. Highly desirable in this sense would be the following property:

$$\models O(X : \gamma_1; \gamma_2) \rightarrow O((X_{11} : \gamma_1); (X_{12} : \gamma_2) + ... + (X_{n1} : \gamma_1); (X_{n2} : \gamma_2)) \ (14)$$
$$\text{where}: \ \ X_{i1}, X_{i2} \subseteq X, \ \ \text{and} \ \ 1 \leq i \leq n$$
$$\text{with} \ \ n = ||\mathcal{P}^+(X) \times \mathcal{P}^+(X)||$$

This formula says that the obligation to execute a (possibly partial) plan, addressed to a group, implies the obligation to choose to perform at least one among all the task allocations, which are possible given that group of agents and that plan. Unfortunately such a formula does not hold unconditionally within our setting, basically for reasons connected with the properties of the $\uplus$ operation (Definition 7): it does not hold in general that if $T_1 \subseteq T_2$ then $T_2 = T_1 \uplus T_2$. Let us now consider a $\Subset$ relation on sets of s-traces defined as follows: $T_1 \Subset T_2$ iff $T_1 \subseteq T_2 \wedge \forall t_3 \in T_2/T_1, \ \nexists t_1 \in T_1 \ \ \text{s.t.} \ \ t_1 \in start(t_3)$. Intuitively, $T_1 \Subset T_2$ iff T_1 and T_2 are in a subset relation and the elements belonging to T_2 but not to T_1 are not sequential extensions of elements of T_1. Considering the $\Subset$ relation, the following can be proved:

Proposition 2. (Choice on task allocations)
If $\forall i \ s.t. \ 1 \leq i \leq n$ and $X_{i1}, X_{i2} \subseteq X, [\![X_{i1} : \gamma_1]\!] \circ [\![X_{i2} : \gamma_2]\!] \Subset [\![X : \gamma_1]\!] \circ [\![X : \gamma_2]\!]$, with $n = ||\mathcal{P}^+(X) \times \mathcal{P}^+(X)||$, then:

$$\models O(X : \gamma_1; \gamma_2) \rightarrow O((X_{11} : \gamma_1); (X_{12} : \gamma_2) + ... + (X_{n1} : \gamma_1); (X_{n2} : \gamma_2)).$$

Proof. Formula (14) follows from two facts: 1) on the ground of Definitions 5 and 7, for each element X_{i1}, X_{i2} we have that $[\![X_{i1} : \gamma_1]\!] \circ [\![X_{i2} : \gamma_2]\!] \Subset [\![X : \gamma_1]\!] \circ [\![X : \gamma_2]\!]$; 2) from Definition 7, holding that $\forall T_1, T_2 \ \text{s.t.} \ T_1 \Subset T_2, T_2 = T_1 \uplus T_2$ we obtain that: $[\![X : \gamma_1]\!] \circ [\![X : \gamma_2]\!] = [\![X_{11} : \gamma_1]\!] \circ [\![X_{12} : \gamma_2]\!] \uplus ... \uplus [\![X_{n1} : \gamma_1]\!] \circ [\![X_{n2} : \gamma_2]\!]$, since $\exists j \ \text{s.t.} \ 1 \leq j \leq n$ and $X_{j1} = X_{j2} = X$.

Despite its complex formulation this property states indeed something remarkably intuitive: (14) holds whenever within a given group what a subgroup does is never sequentially extended, in other words each action performed by a subgroup is always "complete" from a sequential point of view in the sense that no supergroup performs any sequential action longer that the one performed by the subgroup. This property appears to be of relevance also in view of a refinement of the crucial notion of *Step* (Definition 5): once that definition is extended adding this further constraint (14) becomes a validity.

4 Collective Obligations and Coordination as Action: Modeling an Example

In this section we will now show the use of the formalism developed in the previous section to model a collective obligation and show how we can use the validities to derive some individual obligations. We need the properties of the coordinating actions to also make all agents aware of their obligations at the right moment.

Let us consider again the example we have several times touched upon. We have the program committee PC of the DEON'04 workshop, with a chairman c, and the other members $a_1, \ldots, a_n$, such that $PC = \{c, a_1, \ldots, a_n\}$. The collective obligation of the program committee is to notify the authors of the acceptance of their papers: $O(PC : \text{notify})$. It seems reasonable to consider the constraint contained in Proposition 2, which enables formula (14), to be met by the example at issue (each agent perform a sequentially "complete" action), and therefore to be able to infer from $O(PC : \text{notify})$ an obligation on the choice among all the possible task allocations. Let us then suppose that the program committee has selected, out of that choice, the following plan for the notification of acceptance: the chairman collects the submitted papers and divides the papers among the other PC members; the PC members review the papers they have received from the chairman and send their results to the chairman; the chairman makes the final decision which papers are selected for the workshop and informs the authors about the decision. Including the coordination aspect this corresponds with the following task allocation CP, which belongs to the set $PPlan(PC : \text{notify})$[7]:

$$CP = \langle \ \{c\} : \text{collect} \ ; \ \{c\} : \text{coordinate}(c, \text{collect}, \{c\}) \ ; \ \{c\} : \text{divide} \ ;$$
$$\{c\} : \text{coordinate}(c, \text{divide}, \{a_1, ..., a_n\}) \ ;$$
$$(\{a_1\} : \text{review}_1 ; \text{coordinate}(a_1, \text{review}_1, \{c\}))$$
$$\& \ldots \& \ (\{a_n\} : \text{review}_n ; \text{coordinate}(a_n, \text{review}_n, \{c\})) \ ;$$
$$\{c\} : \text{decide} \ ; \ \{c\} : \text{coordinate}(c, \text{decide}, \{c\}) \ ; \ \{c\} : \text{inform\_ authors} \rangle$$

where $skip$ can be substituted for coordinate($\{c\}, \alpha, \{\{c\}\}$). Given the collective obligation $O(G : \text{write})$ and the more concrete obligation on the task allocation sequence CP many consequences, which we here omit for space reasons, can be drawn through validities (2)-(13).

We can now also come back to the question who is taking the initiative for the coordination action. If it is the agent that also performed the action (as in the example above) the coordination becomes an informative action. If the initiative lays with the other party it becomes a checking action. The difference between these two kinds of coordination actions is very important for the issue of responsibility. In the example, the chairman has to inform the other members that he has divided the papers among them. If a member of the PC has not received any paper to review and got no information of the chairman, the

---

[7] See Definitions 1 and 2.

chairman is responsible for the violation of the obligation that some papers are not reviewed. If the coordination action, however, is defined as a checking action: $A : \mathrm{coordinate(c, divide, A)}$, then all the members of the PC have to check whether the chairman has divided the papers, and are therefore also responsible if e.g. the chairman has forgotten to send a member the papers to be reviewed. So, the question who is responsible for a certain action depends on the coordination actions. In a structured group, for example in a formal organization, the coordination can be defined explicitly and therefore also the responsibilities. This is especially necessary, if not everyone in the group has knowledge about the complete plan. In a group which is not so structured it is imaginable that all the members of the group know the plan, and that the coordination consists of informing and checking.

In this example we could assume that every member of the PC has knowledge of the complete plan, which implicitly has the consequence that a member who has not received papers from the chairman, has the obligation to check whether the chairman has done his task. In our formalization we could also indicate that only, e.g., two members a_1 and a_2 have to check whether everyone has received the papers by replacing of $A : \mathrm{coordinate(c, divide, A)}$ by $\{a_1, a_2\} : \mathrm{coordinate(c, divide, A)}$. Of course, it is now up to a_1 and a_2 to make sure that all the PC members know that they have received the papers and have the obligation to review them!

Given the collective obligation $O(PC : \mathrm{notify})$ and the more concrete obligation on the task allocation sequence CP it is now possible to check all kinds of properties of the plan, a certain knowledge about the plan and a coordination scheme with the plan. It becomes possible to check whether all agents know that they have an obligation whenever they have one (true in the example above). Whether all agents know whenever another agent violated his obligation (not true in the example). Whether the coordination is the most efficient possible, given a certain knowledge of the agents (and maybe observable actions), etc. Due to space limitations we omit these discussions in the present paper.

5   Discussion and Conclusions

First of all, there are complex and interesting questions in the logical theory of collective obligation:

> From the point of view of methodological individualism, the action of a collective is in some sense composed of or determined by individual actions performed by the members of the collective. The general study of the nature of that composition, of the dependence of collective actions on individual ones, could be said to constitute the theory of collective action in a narrower sense. ([10])

A first attempt is made to "constitute the theory of collective action in a narrower sense" by the introduction of a plan (task allocation) and coordination.

We discussed collective agency on the basis of some inputs from planning literature in AI in order to provide some definitions of concepts of relevance for our analysis, especially coordination. The notion of coordination given a plan is very useful to determine which agent has to perform and which agent is responsible for a certain sub-action necessary for the fulfillment of the collective action.

We provided a formal framework in which relevant notions for explaining collective agency can be formalized, such as task allocation, collective obligation, and coordination. This framework can help in representing coordination issues to indicate the individual responsibilities, though of a quite simple kind. We believe it provides a valuable basis for further research in collective obligations. First we would like to consider more types of meta-actions. Secondly we would like to check the role of group structures on collective obligations.

References

1. J. Carmo and O. Pacheco. *Deontic and Action Logics for Collective Agency and Roles*. In R. Demolombe and R. Hilpinen, editors, Proc. Fifth International Workshop on Deontic Logic in Computer Science (DEON'00), pages 93-124. ONERA-DGA, 2000.
2. L. Cholvy, Ch. Garion *Collective obligations, commitments and individual obligations: a preliminary study* In Proceedings of 6th international Workshop on Deontic Logic in Computer Science (DEON'02), London, May 2002.
3. L. Cholvy, Ch. Garion *Distribution of Goals Addressed to a Group of Agents*. In Proc. of 2nd Int. Joint. Conf. on Autonomous Agents and Multi-Agents Systems (AAMAS 2003), Melbourne, July 2003.
4. K. S. Decker and V. R. Lesser. *Designing a Family of Coordination Algorithms*. Technical Report No. 94-14, Department of Computer Science, University of Massachussets, Amherst, MA01003, 1995.
5. F. Dignum J.-J.Ch. Meyer. *Negations of Transactions and Their Use in the Specification of Dynamic and Deontic Integrity Constraints*. In M. Kwiatkowska, M.W. Shields, and R.M. Thomas, editors, Semantics for Concurrency, Leicester 1990, pages 61–80, Springer-Verlag, Berlin, 1990.
6. F. Dignum, J.-J.Ch. Meyer, R. Wieringa and R. Kuiper. *A Modal Approach to Intentions, Commitments and Obligations: Intention plus Commitment Yields Obligation*. In M.A. Brown and J. Carmo, editors, Deontic Logic, Agency and Normative Systems (Workshops in Computing), pages 80–97. Springer-Verlag, 1996.
7. F. Dignum and B. van Linder. *Modelling Social Agents: Communication as Action*. In M. Wooldridge J. Muller and N. Jennings, editors, Intelligent Agents III (LNAI-1193), pages 205–218. Springer-Verlag, 1997.
8. J.-J. Ch. Meyer. *A Different Approach to Deontic Logic: Deontic Logic viewed as a Variant of Dynamic Logic*. In Notre Dame Journal of Formal Logic, Volume 29, pages 106–136, 1988.
9. J.-J. Ch. Meyer and W. van der Hoek. *Epistemic Logic for AI and Computer Science*. CUP, 1995.
10. I. Pörn. *The Logic of Power*. Basil Blackwell, Oxford, 1970.
11. L. Royakkers. *Extending Deontic Logic for the Formalization of Legal Rules*. Kluwer Academic Publishers, Dordrecht 1998.
12. L. Royakkers and F. Dignum. *Collective Obligation and Commitment*. In Proceedings of 5th Int. conference on Law in the Information Society, Florence, December, 1998.
13. L. Royakkers and F. Dignum. *No Organization without Obligations: How to Formalize collective obligation?*. In M. Ibrahim, J. Kung and N. Revell, editors, Proceedings of 11th International Conference on Databases and Expert Systems Applications (LNCS-1873), pages 302–311. Springer-Verlag, 2000.
14. S. Russel and P. Norvig. *Artificial Intelligence. A Modern Approach*. Prentice Hall International, 1995.
15. M. Tambe and W. Zhang. *Towards Flexible Teamwork in Persistent Teams: Extended Report*. Journal of Autonomous Agents and Multi-Agent Systems, 3(2):159–183, 2000.