

Group Norms for Multi-Agent Organisations

HUIB ALDEWERELD and VIRGINIA DIGNUM, Delft University of Technology, The Netherlands
WAMBERTO W. VASCONCELOS, University of Aberdeen, United Kingdom

Normative multi-agent systems offer the ability to integrate social and individual factors to provide increased levels of fidelity with respect to modelling social phenomena, such as cooperation, coordination, group decision making, and organization, in both human and artificial agent systems. An important open research issue refers to group norms, i.e. norms that govern groups of agents. Depending on the interpretation, group norms may be intended to affect the group as a whole, each member of a group, or some members of the group. Moreover, upholding group norms may require coordination among the members of the group. We have identified three sets of agents affected by group norms, namely, i) the addressees of the norm, ii) those that will act on it, and iii) those that are responsible for ensuring norm compliance. We present a formalism to represent these, connecting it to a minimalist agent organisation model. We use our formalism to develop a reasoning mechanism which enables agents to identify their position with respect to a group norm, so as to further support agent autonomy and coordination when deciding on possible courses of action.

Categories and Subject Descriptors: I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—*Intelligent Agents; Multiagent Systems*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Norms, norm reasoning, organisation

ACM Reference Format:

H. M. Aldewereld, M. V. Dignum, and W. W. Vasconcelos. 2014. *ACM Trans. Autonom. Adapt. Syst.* V, N, Article A (January YYYY), 32 pages.

DOI: <http://dx.doi.org/10.1145/0000000.0000000>

1. INTRODUCTION

Norms have been used to represent, in compact ways, desirable behaviour that autonomous components should have (alternatively, undesirable behaviour they should not have), so as to provide overall guarantees for distributed, open, and heterogeneous computing solutions. Research on norms has tackled important issues, ranging from logic-theoretic aspects (e.g., [Lomuscio and Sergot 2002]) to more pragmatic concerns (e.g., [García-Camino et al. 2005]).

So far, research on normative multi-agent systems has mostly focused on norms aimed at individuals and has largely ignored a formal treatment of norms aimed at *groups of individuals*. The treatment of norms aimed at groups has been limited to abstractions via the use of *roles* played by the individual to make norms stable over extended periods of time. The core fact, however, remains that while addressing multiple agents singly (namely each agent enacting the role), these norms do not address

Author's addresses: H. M. Aldewereld and M. V. Dignum, Faculty of Technology, Policy and Management, Delft University of Technology, P.O. Box 5015, 2600 GA Delft, The Netherlands, email: {h.m.aldewereld, m.v.dignum}@tudelft.nl; W. W. Vasconcelos, Department of Computing Science, University of Aberdeen, AB24 3UE, United Kingdom, email: wvasconcelos@acm.org.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© YYYY ACM 1556-4665/YYYY/01-ARTA \$15.00

DOI: <http://dx.doi.org/10.1145/0000000.0000000>

these agents together. When addressing a group of agents, aspects such as responsibility and fulfilment play an important role, aspects that are not seen in typical norm representations (since the agent addressed is also responsible, and is also the one to act). In order to grasp the meaning of this difference, we consider for example the obligation for children under the age of 16 to attend school. While the norm addresses children under the age of 16 who are also the ones who must perform the task of going to school, the responsibility and blame lie with their parents/guardians. Stating that “group G should achieve outcome φ ” does not make clear who in the group should actually perform the actions that lead to φ , and who is to blame if the outcome is not achieved. Another example is a removal company, which is obliged by contract to move the contents of someone’s house, including a piano. Given that moving a piano requires special provisions, even though the removal company is the addressee of the obligation, the company is not able to act on it itself and must delegate the task.

This concept of group norms, explicitly differentiating between the groups of agents targeted by the norm, those acting upon it and those responsible for the outcome, creates a set of coordination issues that is not typically seen in individual norms (or typical role-addressing norms). For instance, the agents responsible for the norm have to ensure that they avoid blame, and thus have to ensure that the agents supposed to act upon it are indeed doing what they are supposed to do. The acting agents might need to coordinate whether each of them has to do it, if only one of them has to do it, or if all of them have to do it together (*i.e.*, coordinated group action).

In addition to the coordination issues, group norms present challenges in norm reasoning. Reasoning about norms is of central concern to the regulation or control of the behaviour of a multi-agent system [Boella and van der Torre 2004]. Work on models of norm-governed practical reasoning agents has so far studied the case of norms geared to one agent (or role) [Kollingbaum and Norman 2003], that is, the cases in which the agent is both addressed by and responsible for a norm. In this paper, we present reasoning mechanisms for such group norms. These norms require that the agent is both able to reason about its relation to the norm (*i.e.*, is it addressed, responsible or actor?) and able to coordinate with other agents affected by the norm to determine how to handle the norm.

The contributions of this paper are the following:

- We present a taxonomy of group norms, listing and justifying three types of groups which can be affected by norms, and mapping out a range of possibilities for individual, representative and collective responsibility and actorship.
- We introduce a formal notation to represent group norms, with its underlying semantics, and use this notation to capture the various cases of our taxonomy.
- We formally relate group norms to a simple organisation model, with a view to support normative reasoning.
- We put forward mechanisms to enable agents to reason about group norms.

In Section 2 we present a taxonomy of group norms. This taxonomy provides the motivation for our formalisation introduced in Section 3. We illustrate how our formalisation can be used to represent various situations in Section 4. In Section 5 we present a minimalist model of agent organisations, capturing the concepts of action, role, and power; these provide a structured context within which agents can reason with and about group norms when deciding what to do. In Section 6 we show how agents can use the organisation model to reason about group norms, namely how to act when they are addressed by, are responsible for or are the actors of a norm. In Section 7 we model a cloud computing scenario and illustrate how our approach can help the coordination of agents. We contrast our research with related work in Section 8 and in Section 9 we draw conclusions, raise relevant issues, and indicate avenues for future investigation.

2. A TAXONOMY OF GROUP NORMS

One of the challenges in applying normative theories is that of describing observations and practices using the formal concepts offered by these theories. Our perception is that there is a distinct gap between the normative statements used in everyday practice and those that can be expressed in most (formal) normative modelling frameworks.

In everyday practice, norms can be seen as linguistic statements that prescribe, permit, or oblige actions or outcomes for actors (both individual and collective) [Ostrom 2005]. This definition includes norms of a collective nature, while most formal frameworks only allow the specification of role-oriented individual norms, that is individual norms that are specified in a generic way: *e.g.*, “Program Committee members are obliged to return their reviews before the deadline” or “Project Team members must file a meeting report within 48 hours after the end of the meeting”. While these norms are stated more generically than, *e.g.*, “Agent 0x0FF must return its review before the deadline”, the first example is of a similar individual nature; each agent enacting the program committee role has to return its review(s) before the deadline. The second norm is different, however, since of all agents enacting the ‘Project Team member’ role, only one has to file a meeting report for all agents to be compliant with the norm. In the following we sketch a taxonomy of the different types of group norms, which we then formalise in Section 3.

Before we start to analyse what differentiates the various types of norms shown as examples above and in Section 1, let us first discuss the requirements for a formalism to express group norms. A normative formalisation for the expression of collective/group norms in agent-based organisations should at least include the following elements:

- Deontic modality – necessary to represent and differentiate obligation, prohibition and permission (*how* the norm is influencing behaviours);
- Group representation – necessary to identify those to whom the norm applies (*whom* the norm is influencing);
- Action/state representation – necessary to define what behaviour(s) the norm is influencing (*what* the norm is influencing).

Moreover, the formalisation should comply with the usual desirable properties of any formalisation, in that it should be compact, precise, and machine-processable, and that it should have clear semantics.

Norms provide a way of allocating tasks to groups and individuals, as well as placing restrictions about tasks on groups and individuals. For example, if one wants that agent i should achieve outcome φ , then one can stipulate that “agent i is obliged to bring about φ ”. Likewise, if we want an agent j to refrain from achieving outcome φ , we can state that “agent j is forbidden to achieve φ ”. In the case of an individual agent, interpretation is straightforward, but in the case of a group, things get more complicated.

For instance, by saying that “group G should achieve outcome φ ”, it is not clear who in the group should actually perform the actions that lead to φ and who should be blamed if the outcome is not achieved. We explore in this paper a cloud computing scenario in which group norms govern participants who provide services and managers who assemble solutions from these services – when a service provider fails to deliver its functionality, the manager who is trying to achieve an outcome (stipulated by a norm), by putting together the solution (making use of individual services) may be blamed for norm violation. Depending on the situation, a norm will address different subjects. In many cases, it will address those who must act in order to uphold the norm, whereas in other cases those responsible, but not necessarily acting, are addressed.

Therefore, besides identifying who is addressed by the norm, two other important groups that can affect or be affected by a group norm and must be included in the norm specification, are *i*) responsibility, and *ii*) fulfilment, or actorship. We thus distinguish three types of groups that are affected by a normative statement:

- **Addressees** describe who are addressed by the normative statement; i.e. to whom the normative statement applies; often addressees are either actors or responsables, but other cases may arise;
- **Actors** describe who should achieve (or refrain from achieving) the goal or action of the statement that the norm refers to; and
- **Responsibles** describe who takes care that the norm is upheld, and can be sanctioned if the norm is violated.

In the example about school attendance above, the children are both the addressee and the actor, and the parents are the responsible party. In the piano removal example, the removal company is the addressee and the responsible party, and the subcontractor is the actor. We notice that the concept of addressee is similar to that of *Attribute* in the ADICO grammar for institutions introduced in [Ostrom 2005].

Our approach does not explicitly deal with counterparties of directed norms, establishing obligations from a bearer (responsible for fulfilling the obligation) to a counterparty (normally, some authority or policing body) [Cardoso and Oliveira 2009]. We assume that our norms are *absolute*, that is, they regulate the behaviour of a class of subjects (addressees), instead of regulating the relations between groups. This assumption is realistic as we provide a description of the normative system from the perspective of the counterparty. In our approach, a commitment between addressee and counterparty [Torroni et al. 2009] is a complex inter-relation among addressees, actors and responsables. In our research the addressee represents the subject of the norm from the perspective of the external world, or counterparty. Given that in some cases, the subject of a norm is its actors and, in other cases, its responsables, the explicit use of an addressee makes it easier to represent these two situations.

In our view, *responsibility* expresses who gets the blame when the norm is violated (i.e., when obligations are not acted upon, or when forbidden actions are performed). This aspect of norms was already investigated in [Grossi et al. 2004], with respect to (collective) obligations. It is also similar to the notion of “backward looking” responsibility, as defined in [de Lima et al. 2010]. A limitation in the approaches of [de Lima et al. 2010] and [Grossi et al. 2004] is that they only consider obligations and collective action. Moreover, responsibility can be viewed on several levels with respect to collective norms: *a*) each individual addressed by the norm is responsible for its part in the norm (*individual responsibility*), *b*) the group as a whole is held accountable for failing to comply with a norm (*collective responsibility*), or *c*) a representative of the group is responsible for the failures of the group (*representative responsibility*); that is, a previously appointed member is held accountable for failures of the group to comply with the norm.

Next to responsibility, we differentiate norms in terms of fulfilment or *actorship*, that is, looking at who should fulfil the norm. This is similar to the notion of “forward looking” responsibility, which according to [de Lima et al. 2010] accounts for task allocation and achievement. In some cases, each group member has to do his/her part in fulfilling the norm (that is, *individual actorship*). In other cases, it might be required that the group performs a collective action together (that is, *collective actorship*). While, in most formalisms, these are considered the same (as collective actions are assumed to be decomposable into individual parts), they have a distinct coordination difference. In the latter all the individual parts (namely, the single agent contributes with parts of the collective action) have to be performed in synchrony to be successful, while in

Table I. Taxonomy of group norms, based on responsibility and fulfilment.

	Individual Responsibility	Representative Responsibility	Collective Responsibility
Individual Actorship	(1,1) Individual norm specified in a generic (role-based) way: “PC members are obliged to return their reviews before the deadline.”	(1,2) Individual action, appointed blame: “Employees are obliged to do task, but if task goes wrong blame management.”	(1,3) Individual action, collective blame: “Every child must clean his/her worktable, otherwise the whole class will miss morning break.”
Representative Actorship	(2,1) Appointed action, individual blame: “Group leader must submit report by 12pm, otherwise each student in the group fails the course.”	(2,2) Appointed action, appointed blame: “Every meeting ought to have public minutes. The chairman is responsible for correct minutes being taken by secretary.”	(2,3) Appointed action, collective blame: “The child who broke the flowerpot must confess, otherwise the whole class will miss morning break.”
Collective Actorship	(3,1) Collective action, individual blame: “Groups of more than 3 children are not allowed to enter the shop together.”	(3,2) Collective action, appointed blame: “All PhD students must pack the supervisor’s stuff for the departmental move, otherwise the supervisor will be blamed.”	(3,3) Group action, group blame: “All children must clean the classroom together, otherwise the whole class will miss morning break.”

the former, each agent can decide on its own when to perform the required individual action; one should compare, for instance, lifting a table (which is necessarily done together) with submitting a review (which has less stringent coordination restrictions¹). There is a third case, where it might be that a single group member (or a select subset of the group) fulfilling the norm is sufficient, e.g., as in the filing of a meeting report example mentioned above; we call this latter type *representative actorship*.

This analysis of norms along these two directions, namely, responsibility and fulfilment, can be summarised by the matrix of options shown in Table I, with an example illustrating each case. Note that these examples are meant purely as informal illustrative of the intuitions behind the concept of group norms.

It can be noted that Table I does not explicitly refer to the addressee of each norm. In fact, and as discussed above, in most cases the addressee is either the actor or the responsible group. We introduce the concept of addressee to be able to deal with this difference in a uniform way. An example of a norm where the addressee is the actor is listed in cell (1,1). Cell (2,1) gives an example of a norm where the addressee is the responsible group.

3. GROUP NORMS

Norms provide a compact way of constraining behaviours of groups and individuals. For example, if one wants agent i to achieve outcome φ , then one can stipulate that “agent i is obliged to achieve φ ”, this being formalised as a deontic formula $O_i\varphi$. In the case of an individual agent, interpretation is straightforward, but in the case of a group, many issues arise. By simply stating “group G should achieve φ ”, it is not made explicit who is to act (that is, whether each group member individually, if only one

¹The only coordination aspect of submitting paper reviews is that they are all performed before a particular deadline, instead of their being done simultaneously.

group member, or if all of them together), and who is to blame when violations occur. We can take again the example from the introduction: the obligation for children under the age of 16 to attend school. While the norm *addresses* children under the age of 16, who are also the ones that must *act* upon it, the *responsibility* and blame lie with the parents.

It is with the aim to capture the various stakeholders involved in group norms that we put forward our formalisation below. In the rest of this section we first introduce a language of set definitions required to precisely establish the various groups of our norms. After that we introduce group norms and provide their semantics via temporal logic.

3.1. Set Definitions

We propose to represent groups as set definitions and operations. We assume the existence of a non-empty and finite universal set $Agents = \{ag_1, \dots, ag_n\}$ consisting of the unique identifier of each agent in our society.

Definition 3.1 (Set Definition). A set definition Σ is determined by the following grammar

$$\begin{aligned}\Sigma &::= \Sigma \cup \Sigma \mid \Sigma \cap \Sigma \mid \Sigma \setminus \Sigma \mid \Sigma^C \mid S \\ S &::= \{ag_1, \dots, ag_m\} \mid \{\alpha : P(\alpha)\}\end{aligned}$$

The grammar establishes a language \mathcal{L}_Σ of set definitions and it captures some of the common operations of naïve set theory [Halmos 1960], namely, union, intersection, difference, and absolute complement (with respect to the universal set $Agents$). The S stands for an actual set, and it can be represented as an extensive (finite) listing $\{ag_1, \dots, ag_m\} \subseteq Agents$ of the elements of the set, or an intensional definition $\{\alpha : P(\alpha)\}$, standing for $\forall \alpha \in Agents. P(\alpha)$, that is, all those elements of the universal set that fulfil some property P .

We extend the language of set definitions \mathcal{L}_Σ to represent more sophisticated scenarios. It is common for certain norms to address groups with size restrictions, as in “gatherings of more than 5 people are prohibited”. We can formalise such requirements as $|\Sigma| \circ n$, where \circ is a comparison operator $>, <, \geq, \leq, =$, or \neq and $n \in \mathbb{N}$ (a natural number). These set definitions can be seen as *constrained sets* and they place restrictions on which sets can be built. For instance, if $Agents = \{a, b, c, d\}$ the definition $|\{\alpha : \top\}| = 3$ (where \top stands for “true”, that is a property which is vacuously true for everyone) stands for all subsets of $Agents$ with 3 elements, that is, all groups of 3 agents.

A *set definition* gives rise to different *actual values* of groups, depending on the universal set of agents. Rather than requiring that groups have their individuals listed one by one, our set definitions are more compact and can be re-used for different specific populations of agents. We formally define the value of a set definition Σ with respect to the universal set $Agents$, denoted as $value(\Sigma, Agents) \subseteq Agents$, as follows:

Definition 3.2 (Set Definition Value).

1. $value(\Sigma' \cup \Sigma'', Agents) = value(\Sigma', Agents) \cup value(\Sigma'', Agents)$
2. $value(\Sigma' \cap \Sigma'', Agents) = value(\Sigma', Agents) \cap value(\Sigma'', Agents)$
3. $value(\Sigma' \setminus \Sigma'', Agents) = value(\Sigma', Agents) \setminus value(\Sigma'', Agents)$
4. $value(\Sigma^C, Agents) = Agents \setminus value(\Sigma, Agents)$
5. $value(\{ag_1, \dots, ag_m\}, Agents) = \{ag_1, \dots, ag_m\}$
6. $value(\{\alpha : P(\alpha)\}, Agents) = \{ag_0, \dots, ag_m\}, \forall i, 0 \leq i \leq m, ag_i \in Agents \wedge P(ag_i)$
7. $value(|\Sigma| \circ n, Agents) = value(\Sigma, Agents) \text{ s.t. } |value(\Sigma, Agents)| \circ n$

Cases 1–4 decompose a set definition into its sub-parts, recursively obtaining their values, which then are combined, using the corresponding set operations – this is a straightforward mapping of our notation to the usual semantics of sets. Cases 5 and 6 are the base cases: a set tabulation is itself, and an intensional definition gives rise to every possible sub-set whose elements satisfy property P . Case 7 generically defines the meaning of constrained sets – these are the values of the set definition which satisfy their constraints.

We assume a reference set *Agents* in our discussion, and since we are chiefly interested in what the set definitions actually are, we will simply use the set definitions Σ , meaning $value(\Sigma, Agents)$.

3.2. Group Norms and their Semantics

We formally capture three different groups as set expressions Σ , as introduced in the previous sub-section, as well as the usual components of norms, namely, the deontic modality and the target of the norm [García-Camino et al. 2005; Lomuscio and Sergot 2002].

We use a set of propositions \mathcal{P} , with which one can construct formulae using the usual operators $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$. We represent generic atomic propositions as p, q, r and we use φ, δ, ψ to indicate propositional formulas. The set of well-formed propositional formulas is denoted as $\mathcal{L}_{\mathcal{P}}$. We define group norms as follows:

Definition 3.3 (Group norms). Group norms are of the form ${}^A\mathbf{O}_G^R \varphi < \delta$ (a group obligation) or ${}^A\mathbf{F}_G^R \varphi < \delta$ (a group prohibition), where A, R and G are set definitions (from the language \mathcal{L}_{Σ} of Def. 3.1), and φ, δ are propositional formulae from $\mathcal{L}_{\mathcal{P}}$. We refer to group norms in general as ${}^A\mathbf{D}_G^R \varphi < \delta$ (where \mathbf{D} is either \mathbf{O} or \mathbf{F}).

Intuitively, the annotations A, R and G of the deontic modalities \mathbf{O} and \mathbf{F} correspond to respectively the *actors* (those agents whose behaviours are affected by the norm), those *responsible* for the norm and the *addressees* of the norm. The construct $\varphi < \delta$ informally states “ φ before δ ”, a temporal constraint which enables us to capture deadlines of obligations and periods of prohibitions. It should be noted that we represent achievement obligations, not maintenance obligations. In future work, we will look at the formalisation of group maintenance norms.

The propositions of formula φ may represent actions or properties of states of affairs. In the case of actions, a norm such as ${}^A\mathbf{O}_G^R \text{paint_door} < \delta$ would place an obligation on groups A, G, R to carry out action *paint_door*. More flexibility and expressiveness can be achieved though if φ represents *properties of states*; an example norm is ${}^A\mathbf{O}_G^R \text{painted_door} < \delta$ which stipulates an obligation on groups A, G, R to carry out whatever is required in order to achieve a state in which *painted_door* holds – that is, they should “see to it that” the door is painted, either by painting it themselves or finding someone to paint it on their behalf [Horty 2001]. We do not commit ourselves to either of these options – both can be expressed with our formalism, assuming a suitable semantics for actions and a representation of states is available.

We provide the semantics of our group norms via a temporal logic based on CTL* [Emerson 1990]. Our temporal logical language $\mathcal{L}_{\mathcal{TP}}$ extends our propositional logic $\mathcal{L}_{\mathcal{P}}$ by adding path operators A (all paths), E (some paths), and state operators \bigcirc (next), \square (always), \lozenge (sometime), and \mathcal{U} (until). The language is further enriched with *stit*, *stit*(α, φ) meaning agent α “sees to it that” φ [Belnap and Perloff 1988] and expressing individual action, and *stit*(G, φ) meaning that group G together “sees to it that” φ , for

collective action². The semantics of this logic is constructed in the typical manner from the semantics of CTL* [Emerson 1990] combined with *stit* [Belnap and Perloff 1988]. In our semantics the deontic modalities are handled via an Anderson's reduction [Anderson 1958] of the modality to the reserved $viol(G, A, R, \varphi)$ construct indicating that a violation has happened of G 's norm on φ by (in)action of A under the responsibility of R . We define the meaning of group obligations as follows:

Definition 3.4 (Semantics of Obligation).

$${}^A\mathbf{O}_G^R \varphi < \delta \stackrel{\text{def}}{=} A \left[\Diamond \delta \wedge \left(\neg \delta \wedge \neg stit(A, \varphi) \wedge \neg viol(G, A, R, \varphi) \right) \mathcal{U} \left(\left(\begin{array}{c} \neg \delta \wedge stit(A, \varphi) \wedge \\ \bigcirc(A \Box \neg viol(G, A, R, \varphi)) \end{array} \right) \vee \right. \right. \\ \left. \left. (\delta \wedge viol(G, A, R, \varphi)) \right) \right]$$

Intuitively, this definition expresses that the deadline δ will occur at some point in time and for all paths *either* φ is achieved by the actors ($stit(A, \varphi)$), in which case no violation of the obligation will ever occur ($\bigcirc(A \Box \neg viol(G, A, R, \varphi))$), *or* the state is not achieved, the deadline occurs, and a violation happens ($\delta \wedge viol(G, A, R, \varphi)$). Similarly, we define the meaning of group prohibitions:

Definition 3.5 (Semantics of Prohibition).

$${}^A\mathbf{F}_G^R \varphi < \delta \stackrel{\text{def}}{=} A \left[\left(\begin{array}{c} \neg \delta \wedge \neg stit(A, \varphi) \wedge \\ \neg viol(G, A, R, \varphi) \end{array} \right) \mathcal{U} \left(\left(\begin{array}{c} \neg \delta \wedge stit(A, \varphi) \wedge \\ viol(G, A, R, \varphi) \end{array} \right) \vee \right. \right. \\ \left. \left. (\delta \wedge A \Box \neg viol(G, A, R, \varphi)) \right) \right]$$

Group prohibitions are similar to group obligations, except that the deadline δ is better seen as a deactivation of the prohibition (and may therefore not actually occur in the future states, meaning that the prohibition is not deactivated). So, no violation happens until *either* a violation is triggered by seeing to it that the prohibited state is achieved before the deactivation ($\neg \delta \wedge stit(A, \varphi) \wedge viol(G, A, R, \varphi)$) *or* the prohibition is deactivated (after which no violation can occur ($\delta \wedge A \Box \neg viol(G, A, R, \varphi)$)).

With these definitions of the meaning of group norms, a norm on individual action when $G = A = R$, all referring to a role specification in an organisation, expresses the same as in, for example, [Broersen et al. 2004; Dignum 2004] – all those agents adopting a role (hence belonging to the group) are simultaneously actors, addressees, and responsible parties. For simplification and without loss of generality, in the rest of our discussion we may drop the deadline component of our norms.

To relate the groups and individuals of a norm, we formalise in Section 5 a notion of power (Def. 5.4) – we address *social power* (*viz.*, a relation among individuals of a society, establishing who has authority or control over others [Friedkin 1986]), as opposed to *institutional power* (*viz.*, whereby members of an institution are empowered to perform certain deeds [Demolombe and Louis 2006; Jones and Sergot 1996]). We represent power as a relation $x \preccurlyeq y$ establishing that agent x is under the power of agent y (or conversely, that y has power over x). This relation also applies to groups of agents, as presented later on in the paper.

4. REPRESENTING GROUP NORMS

Using the formal definitions introduced in section 3, we present here the formal specification of the different types of group norms described in Section 2. Depending on the membership of the different groups (addressees, actors, and responsible) the scope of influence of the group norm is different.

²We explain in Section 5.2 how we differentiate collective and individual actions.

Table II. Formal characterisation of the different types of group norms .

	Individual Responsibility	Representative Responsibility	Collective Responsibility
Individual Actorship	$\forall x \in G : \{x\} \mathbf{D}_G^{\{x\}} \varphi$	$A \preccurlyeq R, \forall x \in G : \{x\} \mathbf{D}_G^R \varphi$	$R = G, \forall x \in G : \{x\} \mathbf{D}_G^R \varphi$
Representative Actorship	$A \preccurlyeq G, \forall x \in G : {}^A \mathbf{D}_G^{\{x\}} \varphi$	$A \preccurlyeq G, A \preccurlyeq R : {}^A \mathbf{D}_G^R \varphi$	$A \preccurlyeq G, R = G : {}^A \mathbf{D}_G^R \varphi$
Collective Actorship	$A = G, \forall x \in G : {}^A \mathbf{D}_G^{\{x\}} \varphi$	$A = G, A \preccurlyeq R : {}^A \mathbf{D}_G^R \varphi$	$A = R = G : {}^A \mathbf{D}_G^R \varphi$

Table II provides the formalisation of the different types of group norms described in section 2, as follows. Concerning responsibility, individual responsibility is defined as assigning responsibility to each member of the Addressee set, i.e. norm responsibility is distributed to each of the Addressees, i.e. ${}^A \mathbf{D}_G^R \varphi$ is equivalent to $\bigwedge_{x \in G} {}^A \mathbf{D}_{\{x\}}^R \varphi$. In collective responsibility, the Responsible set R is the same as the Addressee set, $R = G$. Representative responsibility is more complex. As seen in some of the examples above, in some cases those who are responsible for upholding a norm (and getting the blame if not) are not those addressed by the norm (e.g. parents are responsible for the obligation for children to attend school). However, in order to be able to enforce responsibility, a power relation must exist between the Responsible group R and the Actor group A , represented as $A \preccurlyeq R$ (read as “ A is under the power of R ”; a formal definition of power among roles, and hence groups, is put forward in Section 5).

Actorship is dealt with in a similar way. In the individual case, each member of the Addressee set is the Actor of the norm, i.e. ${}^A \mathbf{D}_G^R \varphi$ is equivalent to $\bigwedge_{x \in G} \{x\} \mathbf{D}_{\{x\}}^R \varphi$. In the collective case, all Addressees must act together on the norm, $A = R = G$. Again, in the Representative Actorship case, a power relation is required between the Addressees and the Actors, $A \preccurlyeq G$.

The different interpretations of group norms above also reflect on the target of the norm, i.e. the action being influenced by the norm. In all types of actorship, we assume that the members of the Actor set have the capabilities to act on the target. We are aware that this is a very strong assumption and that the matter of linking capabilities to action is an open research issue which is out of the scope of this paper (we refer the interested reader to, e.g., [Dignum and Dignum 2011]). We interpret the target of the different types of group norms as follows:

- **Individual Actorship** – the target φ is an individual action, i.e. $\forall x \in A : stit(x, \varphi)$
- **Collective Actorship** – the target φ is a collective action of the whole Addressee group, i.e. $stit(G, \varphi)$
- **Representative Actorship** – the target φ can be either an individual or a collective action. That is, if the actor set is singleton, i.e., $A = \{ag\}$, then φ is such that $stit(ag, \varphi)$; if the actor set is composed of more than one agent, i.e. $|A| > 1$, then φ is such that $stit(A, \varphi)$

In the following, we briefly demonstrate the applicability of the formalism by providing the formal representation of the examples described in Table I:

- **Case (1,1)** Individual Responsibility, Individual Actorship – *PC members are obliged to return their reviews before the deadline:*

$$\forall x \in PCMembers : \{x\} \mathbf{O}_{PCMembers}^{\{x\}} return_review < deadline$$

- **Case (1,2)** Representative Responsibility, Individual Actorship – *Employees are obliged to do task, but if task goes wrong blame management:*

$$\forall x \in \text{Employees}, x \preceq \text{Management} : \{x\} \mathbf{O}_{\text{Employees}}^{\text{Management}} \text{task}$$

- **Case (1,3)** Collective Responsibility, Individual Actorship – *Every student must clean his/her worktable, otherwise the whole class will miss the morning break:*

$$\forall x \in \text{Class} : \{x\} \mathbf{O}_{\text{Class}}^{\text{Class}} \text{clean_worktable}$$

- **Case (2,1)** Individual Responsibility, Representative Actorship – *Group leader must submit report by 12pm, otherwise each student in the group fails the course:*

$$\forall x \in \text{Students}, \exists \text{leader} \in \text{Students}, \text{leader} \preceq \text{Students} : \{\text{leader}\} \mathbf{O}_{\text{Students}}^{\{x\}} \text{submit} < 12\text{pm}$$

- **Case (2,2)** Representative Responsibility, Representative Actorship – *Every meeting ought to have public minutes. The chairman is responsible for correct minutes being taken by secretary:*

$$\exists \text{secr} \in \text{Meeting}, \exists \text{chair} \in \text{Meeting}, \text{secr} \preceq \text{chair} : \{\text{secr}\} \mathbf{O}_{\text{Meeting}}^{\{\text{chair}\}} \text{take_minutes}$$

- **Case (2,3)** Collective Responsibility, Representative Actorship – *The student who broke the flowerpot must confess, otherwise the whole class will miss the morning break:*

$$\exists \text{guiltyChild} \in \text{Class}, \text{guiltyChild} \preceq \text{Class} : \{\text{guiltyChild}\} \mathbf{O}_{\text{Class}}^{\text{Class}} \text{confess_break}$$

- **Case (3,1)** Individual Responsibility, Collective Actorship – *Groups of more than 3 children are not allowed to enter the shop together:*

$$\forall x \in \text{Children} : |\{x\} \cup \{y \in \text{Children} : \top\}| > 3 \mathbf{F}_{\text{Children}}^{\{x\}} \text{enter_shop}$$

- **Case (3,2)** Representative Responsibility, Collective Actorship – *All PhD students must pack the supervisor's stuff for the departmental move, otherwise, the supervisor will be blamed:*

$$\text{PhDStudents} \preceq \text{supervisor} : \text{PhDStudents} \mathbf{O}_{\text{PhDStudents}}^{\{\text{supervisor}\}} \text{pack_stuff}$$

- **Case (3,3)** Collective Responsibility, Collective Actorship – *All children must clean the classroom together, otherwise the whole class will miss the morning break:*

$$\text{Class} \mathbf{O}_{\text{Class}}^{\text{Class}} \text{clean_classroom}$$

Exceptional cases where both those acting (A) as well as those responsible (R) differ from those targeted (G), in the most extreme case with no overlap between the actors and the addressees ($G \cap A = \emptyset$) and no overlap between those responsible and the addressees ($G \cap R = \emptyset$), can only happen for group norms with representative actorship and representative responsibility. An example of such a norm is handled in the illustrative scenario given in Section 7.

5. A MINIMALIST ORGANISATION MODEL

There are many approaches to modelling organisations (e.g., [Dignum 2004; Hannoun et al. 2000; McCallum et al. 2008]), but they possess many features in common. Our organisation model aims at capturing only those aspects necessary to explore the phenomena and mechanisms related to group norms and joint behaviour/coordination. We make use of the agent's identity (i.e., the set *Agents*), and we formalise the following aspects:

- *Roles* – these are useful abstractions for (groups of) individuals, conferring generality on organisation specifications. The organisation remains the same, even though different individuals comprise it.
- *Capabilities* – we associate roles with sets of capabilities, represented as actions. These can be understood in two ways: *i*) individuals taking up a role should be able to perform what that role entails; *ii*) they specify what individuals are expected to do in the normal running of the organisation.
- *Power* – within organisations it is necessary to relate roles to one another, so as to facilitate coordination and load-sharing, ultimately enabling objectives (see below) to be achieved. In our proposal, roles exert power (or influence) over other roles, giving rise to power structures such as lines of command, managed teams, hierarchies, and egalitarian teams.

We make use of our propositional language $\mathcal{L}_{\mathcal{P}}$; a set of propositional formulae $\{\varphi_1, \varphi_2, \dots\}$ represents the *conjunction* $\varphi_1 \wedge \varphi_2 \wedge \dots$. We use two special propositions \top and \perp to represent, respectively, “true” and “false”. We assume that the meaning of propositions are captured with formulae establishing logical relations in a knowledge base (or, to use a more modern terminology, a reference ontology) shared by all stakeholders and components³ (e.g., engineers, designers, tools, software agents, and so on). We relate our formulae via logical entailment (formally, “ \models ”) and deduction (formally, “ \vdash ”): for any formulae φ, ψ , if $\varphi \models \psi$ then $\varphi \vdash \psi$ (completeness) and if $\varphi \vdash \psi$ then $\varphi \models \psi$ (correctness).

In particular, in our work we make use of logical implications represented as $(p_1 \wedge \dots \wedge p_n) \rightarrow q$ to forge relationships among propositions, thus providing a background theory (or axioms). We denote as Ω , a set of formulae from $\mathcal{L}_{\mathcal{P}}$, our background theory and we define the meaning of logical implication in terms of entailment as:

$$\text{if } ((p_1 \wedge \dots \wedge p_n) \rightarrow q) \in \Omega \text{ and } \Omega \models p_i, 1 \leq i \leq n, \text{ then } \Omega \models q$$

A similar relation is defined for the “ \vdash ” operator, if we assume its completeness.

We represent a repertoire of actions available during the enactment of an organisation. We propose an idealised representation for actions, and consider these as being *i*) instantaneous (i.e. they take one unit of time to be performed, that is, they do not have a duration or a period for their execution to be completed), *ii*) they are either executed or not (i.e., we do not capture situations whereby actions are partially performed nor do we address scenarios in which actions are performed with degrees of success/quality). We make use of the set \mathcal{P} of propositions as well as a set of negated propositions $\mathcal{P}^- = \{\neg p \mid p \in \mathcal{P}\}$ ⁴.

Definition 5.1 (Action). An action ac is the triple $\langle S, ac, S' \rangle$ where $S \subseteq \mathcal{P} \cup \mathcal{P}^-$, $S' \subseteq \mathcal{P}$, and ac is an action label.

The action labels uniquely identify actions. Our actions model pre-conditions S (a set of possibly negated propositions) which should be satisfied for action ac to be performed, and the result of performing this action (ac ’s post-conditions) is S' , a set of non-negated propositions. We assume a universal, non-empty and finite set of actions

³More realistically, the stakeholders and components have means to relate their knowledge bases (or, to re-phrase this in more modern terms, “align their ontologies”), thus being able to map their knowledge representation on to that of other parties.

⁴It is important to notice that the pre-conditions of an action may contain negated propositions, but not the post-conditions. We present in Section 5.1 an operational semantics showing how agents performing actions update a global state of computation.

$Ac = \{ac_1, \dots, ac_n\}$, such that no two actions have the same label:

$$\forall ac_i \in Ac, \forall ac_j \in Ac. \left[\begin{pmatrix} ac_i = \langle S_i, ac_i, S'_i \rangle \wedge \\ ac_j = \langle S_j, ac_j, S'_j \rangle \wedge \\ ac_i = ac_j \end{pmatrix} \leftrightarrow (S_i = S_j \wedge S'_i = S'_j) \right] \quad (1)$$

The formula above states that within our set of actions, any two elements with the same labels must have the same pre- and post-conditions, *i.e.*, they must be the same. Since actions have unique labels, we shall use ac and ac interchangeably.

Importantly, we model the norms of our scenario in Section 7 addressing *properties of states*, rather than actions. This is without loss of generality since for any action $\langle S, ac, S' \rangle$, where $S' = \{p'_1, \dots, p'_n\}$, we have

$${}^A\mathbf{D}_G^R ac \leftrightarrow {}^A\mathbf{D}_G^R (p'_1 \wedge \dots \wedge p'_n)$$

that is, a norm on an action is equivalent to a norm on its post-conditions. In the case when $S' = \emptyset$, we have

$${}^A\mathbf{D}_G^R ac \leftrightarrow {}^A\mathbf{D}_G^R \top$$

That is, a norm on an action without any effect is equivalent to a norm on the vacuously true proposition “ \top ”, as the empty set is a sub-set of any set, $\emptyset \subseteq S$.

We represent roles as labels available to individual agents when they join the organisation during the enactment. We associate with each role a possibly empty set of action labels, depicting what the role requires to be done:

Definition 5.2 (Role). A role rl is the pair $\langle rl, Ac' \rangle$ where rl is the role label and $Ac' \subseteq Ac$ is a set of action labels (cf. Def. 5.1).

When an agent joins an organisation it takes up one or more roles; by taking up a role the agent agrees to perform any of the actions associated with that role, whenever it is required (or whenever the agent is asked to). We assume a universal, non-empty and finite set of roles $Rl = \{rl_1, \dots, rl_m\}$, such that no two roles have the same label:

$$\forall rl_i \in Rl, \forall rl_j \in Rl. \left[\begin{pmatrix} rl_i = \langle rl_i, Ac_i \rangle \wedge \\ rl_j = \langle rl_j, Ac_j \rangle \wedge \\ rl_i = rl_j \end{pmatrix} \leftrightarrow (Ac_i = Ac_j) \right]$$

That is, any two elements of the set of roles with the same label must have the same sets of actions, thus making these indistinguishable. Because roles have unique names, we shall use rl and rl interchangeably. There are more sophisticated and expressive ways to represent roles, allowing one to define constraints on how many agents can take up the role, the least/highest number of agents for each role, relations among roles (*e.g.*, who takes up roles rl_1, rl_2 should not take up rl_3), and so on, as reported in, for instance, [Dignum 2004; Hannoun et al. 2000; McCallum et al. 2008], but as we aim at a minimalist model, we do not include these here.

When individual agents join organisations they take up roles which they will enact during the life-time of the organisation. We thus consider agents associated with a set of roles, $\langle \alpha, Rl' \rangle$, $\alpha \in Agents$, $Rl' \subseteq Rl$. We define an agent's *capabilities* – the properties of the states that the agent can bring about based on the roles the agent has adopted and the actions associated with these roles:

Definition 5.3 (Capabilities). We define the set of α 's capabilities (when enacting roles Rl'), denoted as $cap(\alpha, Rl')$, as:

$$cap(\alpha, Rl') = \bigcup_{\langle S, ac, S' \rangle \in AllAc} S' \text{ where } AllAc = \bigcup_{\langle rl, Ac \rangle \in Rl'} Ac$$

That is, the capability of an agent α undertaking roles Rl is the union of the post-conditions S' of all actions $AllAc$ of all of α 's roles.

Next, we formally relate roles via *power*, as explored in, for instance, [Jones and Sergot 1996; López y López 2003; Oren et al. 2010], and more recently (and closer to our approach) in [Dignum and Dignum 2011]:

Definition 5.4 (Power). Power $\preceq \subseteq 2^{Rl}$ is a reflexive and transitive relation over the set Rl of roles. If $rl_1 \preceq rl_2$ we say that rl_2 has power over rl_1 or alternatively that rl_1 is under the power of rl_2 .

When an agent enacts a role rl_2 which “has power over” another role rl_1 then that agent may request the help of any agent enacting rl_1 to achieve a particular state of affairs. This request for help is, within a formal organisation, equivalent to *delegation*, since power relations should be followed without question. Power and delegation is best understood via the “see to it” (*stit*) operator [Horty 2001], $stit(rl, \varphi)$ standing for “role rl sees to it that φ ”.

If $stit(rl, \varphi)$ and $rl = \langle rl, Ac' \rangle$ hold, then one of the following properties must also hold:

- (1) Role rl has associated actions with combined post-conditions logically entailing φ . Formally:

$$S^* = \left(\bigcup_{\langle S, ac, S' \rangle \in Ac'} S' \right) \text{ and } S^* \models \varphi$$

- (2) Role rl has power over roles rl'_i each of which can see to it that φ'_i , and these combined φ'_i logically entail φ . Formally,

$$\Phi' = \{\varphi'_i : rl'_i \preceq rl \wedge stit(rl'_i, \varphi'_i)\} \text{ and } \Phi' \models \varphi$$

Since the power relation is reflexive, that is, all roles have power over themselves ($\forall rl \in Rl. rl \preceq rl$), then property 2 above also addresses scenarios in which agents delegate responsibility over the achievement of some φ'_i but they also retain responsibility for achieving some φ'_i , through their own actions.

The power relation can be extended to relate individual agents: let there be two agents ag_1, ag_2 with associated sets of roles $\langle ag_1, Rl_1 \rangle, \langle ag_2, Rl_2 \rangle$; if there is a role $rl'' \in Rl_2$ for which there is a role $rl' \in Rl_1$ such that $rl' \preceq rl''$, then we say $ag_1 \preceq ag_2$. That is, ag_2 has power over ag_1 if at least one of ag_2 's roles has power over one of ag_1 's roles. We notice that this is a “weak” definition of power which could, in some situations, lead to loops in delegation – this is an undesirable feature of an organisation specification to which designers should be alerted⁵. A stronger definition would require that, in addition to the requirements above, we also had $rl'' \not\preceq rl'$, for all roles rl', rl'' .

We further extend the power relation to account for groups (sets) of agents, as follows:

Definition 5.5. Given sets $Agents_1, Agents_2 \subseteq Agents$, and a power relation $\preceq \subseteq 2^{Agents}$ we say that $Agents_2$ has power over $Agents_1$, denoted as $Agents_1 \preceq Agents_2$, if,

⁵During the enactment of an organisation (run-time) each agent adopts a sub-set of roles. If the power relation has *any loop* then there is *potential* for loops when agents (acting in different roles) are delegating. By detecting/flagging loops in the power relation at design time we are warning designers about such potential loops in delegation at run-time. More sophisticated representations for roles [Dignum and Dignum 2011; Carmo and Pacheco 2001; Pacheco and Carmo 2003] addressing features such as “at most one agent should be in this role” (cardinality of a role) and “whomever takes up this role cannot take up this other role” (compatibility of roles), could avoid certain combinations of roles, thus partitioning the graph of roles (vertices) and power relations (edges) into sub-graphs without loops.

and only if

$$\forall \alpha' \in Agents_1, \exists \alpha'' \in Agents_2 : \alpha' \preceq \alpha''$$

That is, every member of $Agents_1$ is under the power of at least one member of $Agents_2$.

In order to model realistic scenarios, group norms ${}^A\mathbf{D}_G^R \varphi$ (where \mathbf{D} is either \mathbf{O} or \mathbf{F}) should fulfil the following properties:

- (1) $A \preceq R$ – the group of actors A must be under the power of the responsible group R . This property ensures that those responsible for the norm should be able to delegate to actors.
- (2) $A \preceq G$ – the group of actors A must be under the power of the group G addressed by the norm. This property ensures that addressees are also able to delegate to actors.

Both properties above can be checked at run-time, when the groups are instantiated with specific members. Concerning the power relation between Addressees and Responsibles, in most realistic models, it will be the case that $G \cap (A \cup R) \neq \emptyset$, and $G \cap A \neq \emptyset \vee G \cap R \neq \emptyset$, i.e. the Addressee group overlaps in some extent with the Actors or Responsibles group. We consider the specification of these relations for a given application domain to be design decisions, and therefore do not impose $G \preceq R$ nor $R \preceq G$. (E.g., a norm aimed at a group of junior engineers G has a senior manager responsible R for it, and operators as actors A) R and G do not directly relate power-wise.

5.1. A Computational Model for Norm-Aware Multi-Agent Organisations

We outline a computational model for norm-aware multi-agent organisations, providing a context for Defs. 5.1–5.5. Our model is built around an explicit representation of the *global state* $S^* \subseteq \mathcal{P}$ of the computation in which pre-conditions of actions are checked for and their post-conditions (effects) are recorded: given a state S_i and an action $\langle S, ac, S' \rangle$, if S holds in S_i (see below) then the action can be applied and we obtain a next state⁶ $S_{i+1} = S'$.

We follow the architecture for distributed norm management proposed in [García-Camino et al. 2008; Vasconcelos et al. 2012], and consider a global state which is updated as a result of individual agent's actions – this is similar to *transition systems* [Emerson 1990]. We represent this as the construct $\boxed{S_0} \Rightarrow \boxed{S_1} \Rightarrow \dots$, showing a sequence of global states $\langle S_0, S_1, \dots \rangle$ created from an initial state S_0 with the “ \Rightarrow ” operation indicating the application of a set of actions (from possibly many individual agents) on S_i , giving rise to S_{i+1} .

Negated propositions in pre-conditions of actions (cf. Def. 5.1) are interpreted as *negation as failure* [Clark 1978], that is, they hold if they cannot be proven true. Since we are dealing with single propositions (rather than formulae), to check if a negated proposition $\neg p$ holds in a state S we need to check that p does not appear in S , that is, $S \models \neg p$ if, and only if, $p \notin S$. Given a set $S' \subseteq \mathcal{P} \cup \mathcal{P}^-$ of (possibly negated) propositions, we establish when $S \models S'$:

$$S \models S' \text{ if, and only if, } \begin{cases} \forall (\neg p) \in S', p \notin S \\ \forall p \in S', p \in S \end{cases}$$

⁶This means that propositions are not implicitly recorded in (copied onto) the next state; a proposition will only be copied from one state onto the next state if it appears both in the pre- and post-conditions of an action.

Given a set of actions Ac and a global state S^* , individual agents can compute the subset of actions $Ac' \subseteq Ac$ whose pre-conditions hold in S^* :

$$applicableActions(Ac, S^*) = \{\langle S, ac, S' \rangle \in Ac \mid S^* \models S\}$$

An underlying infrastructure controls access to global states, and mediates how and when an update is to take place. Agents decide on the actions they want to perform (chosen from *applicableActions*), and update the global state directly. Since the post-conditions of actions do not contain negated propositions (cf. Def. 5.1) the resulting state will not have inconsistencies⁷. Agents join an organisation and take up roles, each of which with its own associated actions (cf. Def 5.2). Given a role $rl \in Rl$, $rl = \langle rl, Ac' \rangle$, then the agent can compute *applicableActions*(Ac', S^*).

We provide means to check which group norms hold, using their deadlines/periods δ . For simplicity our norms do not have activation conditions, and this is interpreted as a norm being active until δ holds in S^* , that is,

$$active({}^A D_G^R \varphi < \delta, S^*) \text{ holds if, and only if, } S^* \models \delta$$

Agents are able to find out all those active group norms, and use the group norm reasoning mechanism (Alg. 1) defined in Section 6 to establish how to handle the group norm – as a member of a group of actors, addressees or responsables.

5.2. Individual and Collective Actions

Our formalisation in Def. 5.1 caters for both individual and collective actions. We differentiate between these in a pragmatic fashion: collective actions are those whose post-conditions (effects) are achievable via the combination of other (individual or collective) actions. Formally, given a set of actions Ac , $ac \in Ac$, $ac = \langle S, ac, S' \rangle$ is a collective action if, and only if, the conditions below hold

- (1) for some $n \geq 2$, there are $ac_i \in Ac$, $ac_i \neq ac$, $ac_i = \langle S_i, ac_i, S'_i \rangle$, $1 \leq i \leq n$, that is, there are (at least two) other actions ac_1, \dots, ac_n ,
- (2) $S \models \bigcup_{i=1}^n S_i$, whose pre-conditions are entailed by S
- (3) $(\bigcup_{i=1}^n S'_i) \models S'$, whose post-conditions entail S'
- (4) Given a set *AssocRl* of pairs $\langle \alpha, Rl' \rangle$, $\alpha \in Agents$, $Rl' \subseteq Rl$, establishing the roles Rl' which individual agents α are enacting, then $S' \not\subseteq cap(\alpha, Rl')$ for all $\langle \alpha, Rl' \rangle \in AssocRl$.

The first condition establishes the “break-down” of a collective action into other actions⁸. The second condition ensures that actions ac_i are applicable whenever ac is applicable. The third condition ensures that the combined effect (post-conditions) of actions ac_i addresses all post-conditions of ac . The fourth condition states that for a particular organisation enactment (that is, agents associated with specific roles), a collective action is not within the capabilities of any one individual agent α however many roles Rl' it has adopted.

⁷There could be situations in which, although there are no occurrences of p and $\neg p$, the domain modelled could establish that propositions *lightOn* and *lightOff* are contradictory. In such circumstances, a more sophisticated global process would ensure that actions are handled in some order (say, starting with the agents with highest power) and any action which could introduce a contradiction would be rejected.

⁸Collective actions have large sets of post-conditions reflecting the “effort” to be spent in order to achieve them. By breaking apart an action into other (simpler) actions which together achieve the same effects (post-conditions) we capture the delegation process supported by the power relation, and which is explored in our reasoning mechanisms.

The assumption made in connection to Def. 5.1, represented as Formula 1, also implies⁹ that no two different actions have the same pre- and post-conditions. This means that our set of actions gives rise to a tree-like structure whereby a parent node is an action whose branches are other simpler actions, as illustrated in Figure 1. Any of the

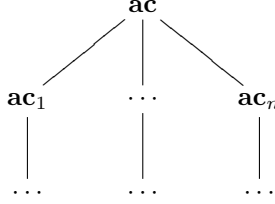


Fig. 1. Break-down of action ac into actions ac_i , $1 \leq i \leq n$

actions ac_i may be decomposed into other simpler actions; this decomposition is similar to the modelling of Hierarchical Task Networks in planning systems [Ghallab et al. 2004].

When an obliged action cannot be achieved by any one single agent (under its many roles) in an organisation enactment, then the action is deemed collective and it should be “farmed out” to groups of agents so that, by joining their capabilities, the collective action can be achieved and the obligation fulfilled. On the other hand, an individual action is within the capabilities of a single agent in an organisation; formally, $ac = \langle S, ac, S' \rangle$ is an individual action if, and only if, $S' \subseteq cap(\alpha, Rl')$ for some $\langle \alpha, Rl' \rangle \in AssocRl$.

Collective actions are carried out via sets of individual actions. The effects of individual actions should thus be combined together to represent the collective actions they achieve. We capture this via *domain axioms*: a collective action $ac = \langle S, ac, \{p'_1, \dots, p'_r\} \rangle$ is achievable via individual actions $\langle S_i, ac_i, \{q'_{[1,i]}, \dots, q'_{[m_i,i]}\} \rangle$, $1 \leq i \leq n$, if, and only if, $(\bigwedge_{i=1}^n \bigwedge_{j=1}^{m_i} q'_{[j,i]}) \rightarrow (\bigwedge_{k=1}^r p_k)$ is a domain axiom¹⁰. Domain axioms enables agents to reason about how an action can be decomposed without having to carry out an extensive search for every possible combination of individual actions. Knowledge engineers supply domain axioms and these will also be used to refine the records of actions in global states – records of individual actions which were carried out will give rise to the record of a collective action. We observe that the overall effect of collective actions, individual actions and domain axioms is similar to “counts-as” rules [Grossi and Jones 2013], relating brute facts (individual actions) with institutional facts (collective actions).

5.3. Coordination and Group Norms

Agents must coordinate their activities factoring in their roles (with associated capabilities), their membership to groups, and active norms which are applicable to the groups. Our reasoning mechanisms (Algorithms 2 and 3) introduced in the next section

⁹Formula 1 uses a bi-conditional, so if the pre- and post-conditions of two actions are respectively the same, then the two actions are the same.

¹⁰If we regard the action label ac as a shorthand to the post-conditions S' of action $ac = \langle S, ac, S' \rangle$, then domain axioms can be made more compact. In our case above, we would have the domain axiom $(\bigwedge_{i=1}^n ac_i) \rightarrow ac$ which could be used by agents when reasoning about how a collective action is to be broken up and its sub-actions delegated.

make use of two procedures *coordinate* and *coordinate'* to support coordination among agents, which we explain below.

Procedure *coordinate* is invoked by an agent α who is attempting to coordinate with agents $Agents'$ in order to achieve φ , and α is willing to contribute with S' :

$$coordinate(\alpha, Agents', contribute, S', \varphi)$$

We do not prescribe any solutions to the coordination mechanism itself, that is, the messaging/network topology, actual contents of messages and their order, or any guarantees such mechanism should have – research on distributed coordination [Williams 2014] and planning [Han et al. 2014] provides candidate solutions for this. The coordination process will go through a series of rounds whereby agents $\alpha' \in Agents'$ receive requests from α to help with achieving φ ; the request may also include (partial) information on α 's contribution S' . Agents will reply to requests offering their own contributions, as part of their deliberation process about what to do next, also factoring in other group norms – research on how deliberation can be extended with normative considerations is reported in, for instance, [Meneguzzi et al. 2015]. Agent α will select who should contribute with what and consider whether it is necessary to send more requests, in case the responses so far are not sufficient to achieve φ . After a finite number of rounds the coordination process may succeed (and the procedure returns \top , that is, the Boolean value *true*) or may fail (and the procedure returns \perp , the Boolean value *false*).

The same *coordinate* procedure supports agents when they are coordinating about who should *refrain* from carrying out actions. Such coordination is needed when actions executed individually by members of a group will cause a violation of a group prohibition. In such circumstances, individual actions by themselves are not forbidden, but their combined effects amount to a forbidden collective action being carried out – for instance, although individual agents are not forbidden to be in a lift, if more than, say, 8 of them are in the lift, then a group prohibition is violated. When an agent is deliberating about which action it should carry out next, then it should factor in coordination requests to refrain from performing actions which could contribute to a violation of a group prohibition.

The other procedure *coordinate'* used in our reasoning mechanisms is invoked by an agent α attempting to coordinate with those agents $Agents'$ who belong to the R group of agents responsible for group norm ${}^A\mathbf{D}_G^R \varphi$:

$$coordinate'(\alpha, Agents', {}^A\mathbf{D}_G^R \varphi)$$

The procedure returns the pair $\langle \alpha', {}^{A'}\mathbf{X}_{G'}^{R'} \phi \rangle$ with the outcome of the coordination: $\alpha' \in (Agents' \cup \{\alpha\})$ has agreed (as a member of the group responsible for the norm) to be in charge of ${}^{A'}\mathbf{X}_{G'}^{R'} \phi$. The coordination process may convert the original norm ${}^A\mathbf{D}_G^R \varphi$ into an altogether different norm ${}^{A'}\mathbf{X}_{G'}^{R'} \phi$ (a special case is when ${}^{A'}\mathbf{X}_{G'}^{R'} \phi = {}^A\mathbf{D}_G^R \varphi$, that is, the coordination preserves the original norm). This conversion would allow, for instance, group norm ${}^A\mathbf{O}_G^{\{ag_1, ag_2\}} (p \wedge q)$ to become norms ${}^A\mathbf{O}_G^{\{ag_1\}} p$ and ${}^A\mathbf{O}_G^{\{ag_2\}} q$, that is, the group $\{ag_1, ag_2\}$ of agents responsible for the norm have agreed to take responsibility over parts of the original group norm. Another important conversion would make use of domain axioms to work out how a group norm on the effects of a collective action (cf. Section 5.2) could be split into distinct group norms (with potentially different groups) over the effects of other (collective) actions.

We do not detail the *coordinate'* process here either, but many of the considerations raised for *coordinate* also apply here. Additionally, the conversion of group norms

amounts to deontic reasoning added with the manipulation of groups: we do not provide a solution to this, and we are aware of the computational complexity involved in solving this in general. Our intention here is to establish when such a process will be invoked within normative group reasoning, define its information requirements in terms of input and outputs, and provide an initial list of requirements for solutions to this process. Although the conversion of group norms in general is computationally demanding and semi-decidable [Horty 2001; McNamara 2006], the process may target simpler solutions – in particular, *coordinate'* may simply assign someone in R to be responsible for the norm, or use a domain axiom to break up a norm on a collective action onto a norm (or norms) on other actions (collective or individual ones).

6. REASONING ABOUT GROUP NORMS

The objectives of an organisation can only be realised when agents take up the roles described in the organisation definition. We assume that agents have their own motivations to decide on which roles they will take on, but once a role enactment is fixed, the agent is able to act on the capabilities described for its role(s). Moreover, agents have access to the organisation specification consisting of:

- The set of agents *Agents* enacting/joining the organisation.
- The set of actions Ac (Def. 5.1), the set of roles Rl (Def. 5.2) and their associated capabilities (Def. 5.3).
- A set *AssocRl* of pairs $\langle \alpha, Rl' \rangle, \alpha \in Agents, Rl' \subseteq Rl$, recording which roles Rl' individual agents α are enacting (hence formally associated with).
- The power relation (Defs. 5.4–5.5) among roles and sets of agents (enacting roles).

The organisation specification allows agents to figure out each other's (as well as their own) roles, capabilities, and who has power over whom.

We furthermore assume an open environment in which heterogeneous agents, possibly developed by third parties, may join the organisation. This means that role enactment can take many forms, *i.e.*, depending on the agent's own "personality", its interpretation of what is expected from it as enactor of the role (and how to decide about its role norms) may vary. For instance, an agent with a strong sense of responsibility will first consider the norms for which it belongs to the Responsible group, whereas an agent that has a strong sense of duty may start by considering the norms for which it is an Actor. In the following, we describe, in pseudo-code, reasoning mechanisms for role enacting agents.

We initially present in Algorithm 1 the general reasoning mechanism, consisting

Algorithm: *groupNormReasoning*($\alpha, {}^A\mathbf{D}_G^R \varphi$)
if $\alpha \in \text{value}(G, Agents)$ **then** *addressment*($\alpha, {}^A\mathbf{D}_G^R \varphi$);
if $\alpha \in \text{value}(R, Agents)$ **then** *responsibility*($\alpha, {}^A\mathbf{D}_G^R \varphi$);
if $\alpha \in \text{value}(A, Agents)$ **then** *actorship*($\alpha, {}^A\mathbf{D}_G^R \varphi$);

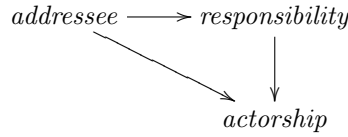
Algorithm 1: Group norm reasoning

of an assessment of the value of the norm groups and a check whether or not the agent belongs to these. Depending on which group the agent belongs to, separate sub-mechanisms are invoked, and these are explained in the remainder of this section. We assume that the mechanisms have access to a global set *Agents* comprising the organisation, as well as the specific actions, roles, capabilities and (group) power relations (cf. Defs. 5.1–5.5).

Input parameters α and ${}^A\mathbf{D}_G^R \varphi$ stand for, respectively, the agent's identity and a group norm under consideration¹¹. We order the agent's considerations about group membership: it first checks if it is part of the group of addressees of the norm, then if it is part of the group of agents responsible for the norm, and finally the agent checks if it is an actor of the norm. If none of these situations arise, then the agent does not have to factor in the group norm in its decision. This ordering is due to the relationships among agents belonging to the distinct groups: the addressee analysis may require responsibility and actorship analysis (depending on the circumstances); the responsibility analysis may require actorship analysis, but actorship is self-contained. These mechanisms are described in the rest of this section.

The mechanism above also caters for situations in which agents simultaneously belong to more than one of the groups A , R , or G . As we show below, agents in G that are addressed by the norm will “farm out” the norm among those responsible (in group R) and those acting (in group A); those agents responsible for the norm (in group R) will require the help of acting agents A . When an agent is part of more than one group, then we will have the phenomenon of agents calling upon themselves to handle the norm under a different guise.

Our group norm representation is used in mechanisms to support agents reasoning about actorship (Algorithm 2), responsibility (Algorithm 3), and addressment (Algorithm 4). We illustrate their interdependence as



The reasoning invokes individual planning (during actorship reasoning), group coordination (during responsibility reasoning) and individual deliberation (during addressee reasoning). The reasoning is interleaved with message-passing (protocols) to enable coordination, as well as communication regarding who is taking up actorship, and to signal which norm has been violated.

6.1. Reasoning about Actorship

Group norms are ultimately “processed” by actors: these are agents belonging to the group A of norms ${}^A\mathbf{D}_G^R \varphi$ and their behaviours should be affected by these norms. We recall that our group norms consider propositional formulae φ (cf. Def. 3.3), and we note that these can come about as a result of a coordinated joint action among various agents, each contributing some effort to achieve or avoid φ . We propose the reasoning mechanism depicted in Algorithm 2 to enable norm-aware decision-making and coordination among acting agents. Line 1 computes all those sub-groups of actors whose capabilities (under their respective adopted roles in the organisation) when pooled together logically derive (or entail) φ ¹². Line 2 computes those coalitions to which α , the agent executing the algorithm, belongs. Lines 3–11 describe the provisions for norm-compliant behaviours. Line 12 is a place holder for non-norm-compliant behaviours – these might include, for instance, having α alerting other team-members (that is, all

¹¹For simplicity, we omit deadlines/periods of norms in our mechanisms, and the assumption is that the input norm is currently active, that is, its deadline/period has not expired and hence it must be considered. This assumption can be relaxed, but all algorithms should initially check whether or not the norm is still active.

¹²This amounts to finding *all* minimal *coalition* of agents who can achieve φ collectively [Ågotnes and Alechina 2011].

Algorithm: $actorship(\alpha, {}^A D_G^R \varphi)$

```

1 let  $\mathcal{A} = \{Agents'_0, \dots, Agents'_n\}$ , where each  $Agents'_i \subseteq value(A, Agents)$  is the
   smallest set s.t.  $\forall \alpha' \in Agents'_i. \langle \alpha', Rl' \rangle \in AssocRl \wedge (\bigcup cap(\alpha', Rl')) \vdash \varphi$ 
2  $\mathcal{A}^\alpha \leftarrow \{Agents'_i \in \mathcal{A} \mid \alpha \in Agents'_i\}$ 
3 if  $complyNorm({}^A D_G^R \varphi, \mathcal{A}^\alpha)$  then
4    $MyCap \leftarrow cap(\alpha, Rl)$  where  $\langle \alpha, Rl \rangle \in AssocRl$ 
5   for  $Agents' \in \mathcal{A}^\alpha$  do
6     if  $D = O$  then
7       if  $coordinate(\alpha, (Agents' \setminus \{\alpha\}), contribute, MyCap, \varphi)$  then
8         return  $\top$ 
9       else
10        if  $\neg coordinate(\alpha, (Agents' \setminus \{\alpha\}), refrain, MyCap, \varphi)$  then
11          return  $\perp$ 
12      if  $D = O$  then
13        return  $\perp$ 
14      else
15        return  $\top$ 
16    else
17      ... non-norm-compliant behaviour...

```

Algorithm 2: Reasoning about actorship

α' such that $\alpha' \preceq \alpha$ and $\alpha \preceq \alpha'$) or informing its in-line manager (that is, an α' such that $\alpha \preceq \alpha'$ and $\alpha' \not\preceq \alpha$) about its decision to not comply with a norm. We focus on norm-compliant behaviour: line 4 computes agent α 's own capabilities $MyCap$ within the organisation, based on its roles.

Lines 5–13 loop through each minimal coalition to which α belongs, checking whether the norm is an obligation (line 6) or a prohibition (line 9). In the case of an obligation, α tries to coordinate with the coalition $Agents'$ to *contribute* with its capabilities $MyCap$ to achieve φ – it is enough for one such coordination attempt to succeed for the actorship algorithm to return \top (line 8). In the case of a prohibition (line 9), α attempts to coordinate with $Agents'$ to agree on who is to refrain from doing what in order to not achieve φ (and hence abide by the prohibition) – it is enough for one coordination attempt to fail (that is, $\neg coordinate$ holds in line 9) for the actorship algorithm to also fail. In both cases, the loop is cut short and a result is returned.

If, however, the loop in lines 5–10 explores all coalitions without returning anything, then the test in lines 11–13 confirms that agent α was unsuccessful in coordinating to fulfill the obligation (line 12) or α was successful in coordinating to abide by a prohibition (line 13), otherwise the commands in line 8 (respectively, line 10) would have been performed and the flow of execution would never have reached line 12 (respectively, line 13).

We use logical implications to capture domain axioms, as explained previously, and we assume completeness of the “ \vdash ” operator (which appears in line 1 of Alg. 2).

6.2. Reasoning about Responsibility

Agents belonging to group R of a norm ${}^A D_G^R \varphi$ are responsible for the norm, that is, they are to blame if the norm is violated. Those agents responsible enlist the help of acting agents belonging to the group A of our norms. However, agents responsible for the norm need to agree among themselves who will take the initiative to contact the actors. Moreover, the agents responsible for the norm should only contact actors over

whom they have power. This process is represented in Algorithm 3. Line 1 invokes a

Algorithm: $\text{responsibility}(\alpha, {}^A\mathbf{D}_G^R \varphi)$

```

1  $\langle \text{inCharge}, {}^{A'}\mathbf{X}_{G'}^{R'} \phi \rangle \leftarrow \text{coordinate}'(\alpha, (\text{value}(R, \text{Agents}) \setminus \{\alpha\}), {}^A\mathbf{D}_G^R \varphi)$ 
2 if  $\text{inCharge} = \alpha$  then
3    $\text{ActorsSet} \leftarrow \{\alpha' \mid \alpha' \in \text{value}(A', \text{Agents}) \wedge \alpha' \preceq \alpha\}$ 
4   for  $\alpha' \in \text{ActorsSet}$  do
5     if  $\text{actorship}(\alpha', {}^{A'}\mathbf{X}_{G'}^{R'} \phi)$  then
6       return  $\top$ 
7   return  $\perp$ 

```

Algorithm 3: Reasoning about Responsibility

coordination mechanism whereby α interacts with the other members of R regarding who should be ultimately responsible for the norm. This process should factor in the nature of φ – it might be the case that more than one agent should become involved in procuring actors to fulfill the norm. We note that the result of this coordination exercise could be the re-casting of the original norm into distinct norms whose overall effect, when they are complied with, is the same as the original norm¹³ and we indicate this in the algorithm with a (possibly) different norm ${}^{A'}\mathbf{X}_{G'}^{R'} \phi$ being agreed to (line 2) by the group R to have α being in charge.

Step 3 computes the set of actors α' over which α has power ($\alpha' \preceq \alpha$). Line 4 establishes a loop over all actors, repeatedly invoking the actorship reasoning mechanism of algorithm 2, stopping (and returning “ \top ”, that is, success) when the first of the acting agents handles the group norm. Otherwise, when we run out of choices for acting agents, the mechanism reports a failure “ \perp ”.

6.3. Reasoning about Addressment

We finally consider the case when an agent is a member of the addressed group G of norm ${}^A\mathbf{D}_G^R \varphi$, depicted in Algorithm 4. In this case, the mechanism computes (line 1) the set of agents α' responsible for the norm, and over which α has power ($\alpha' \preceq \alpha$). Line 2 starts a loop invoking, for each α' , the *responsibility* mechanism depicted in algorithm 3, stopping when the first agent handles the norm. Lines 5–9 explores the exception to the responsibility mechanism, that is, a member α of the addressed group A , directly takes responsibility over finding actors to deal with the norm – this part of the mechanism corresponds to lines 3–7 of the responsibility mechanism.

6.4. Discussion

Our representation of group norms caters for three distinct groups involved. Being able to differentiate among those *addressed* by the norm (*i.e.*, group G), those *responsible* for the norm (*i.e.*, group R), and those *acting* on the norm (*i.e.*, group A), allows us to formally capture interesting and realistic situations. For instance, a norm such as “anyone under the age of 16 is obliged to attend school”, can be represented as ${}^A\mathbf{O}_G^R \text{attendSchool}$ where

— R is $\{x : x \in \text{People} \wedge \text{parent}(x, y) \wedge \text{under16}(y)\}$, that is, the group responsible for the norm consists of anyone who is a parent of an under-16 person;

¹³We illustrate this with a norm (without the groups) $\mathbf{OLiftTable}$ and axiom $(\text{liftEndA} \wedge \text{liftEndB}) \leftrightarrow \text{liftTable}$, which gives rise to $\mathbf{OLiftEndA} \wedge \mathbf{OLiftEndB}$.

Algorithm: $addressment(\alpha, {}^A\mathbf{D}_G^R \varphi)$

```

1  $ResponsibleSet \leftarrow \{\alpha' \in value(R, Agents) \wedge \alpha' \preceq \alpha\}$ 
2 for  $\alpha' \in ResponsibleSet$  do
3   if  $responsibility(\alpha', {}^A\mathbf{D}_G^R \varphi)$  then
4     return  $\top$ 
5  $ActorSet \leftarrow \{\alpha' \in value(A, Agents) \wedge \alpha' \preceq \alpha\}$ 
6 for  $\alpha' \in ActorSet$  do
7   if  $actorship(\alpha', {}^A\mathbf{D}_G^R \varphi)$  then
8     return  $\top$ 
9 return  $\perp$ 

```

Algorithm 4: Reasoning about Addressment

— $G = A$ and they are $|\{y : y \in People \wedge under16(y)\}| = 1$, that is, those addressed and the actors are individuals (*i.e.*, sets of size one) who are under-16.

The norm “groups of more than 3 children are forbidden to be in a shop” is formalised as ${}^A\mathbf{F}_G^R inShop$ where

- G is *Children*, that is, the norm is addressed at all kids;
- R is $|\{x : x \in Children\}| = 1$, that is, each kid is individually responsible for the norm (hence the set has exactly one member);
- A is $|Children'| > 3$, that is, the actors are all groups of kids with 3 or more members

A third example is the norm “the chairperson of a meeting is obliged to have the secretary circulating the minutes”, formalised as ${}^A\mathbf{O}_G^R circulateMinutes$, where

- $G = Meeting$, that is, the norm is addressed to all those attending the meeting.
- $R = \{chair\}$, that is, the responsible for the norm is the chairperson (singleton set);
- $A = \{secretary\}$, that is, the secretary (a singleton set) is the one acting on the norm.

Our group norm representation has been put to use in mechanisms to support agents reasoning about actorship (Algorithm 2), responsibility (Algorithm 3), and addressment (Algorithm 4). Completeness is achieved as the net effect of our mechanism is that addressee agents exhaustively try to find someone responsible or someone to act (invoking responsibility and actorship analyses), the responsible agents exhaustively try to find actors, and finally the actors try to plan, factoring in the constraints of the norm (avoiding prohibited states, and aiming at obliged states). Termination of the process is guaranteed if there are no loops in the power relation, as all groups are finite, and so are the agents’ individual roles and actions, and the interaction (although not shown) converges with a successful action/plan or a message declining to help. The complexity of the three combined analyses, in the worst case, is the permutation of the elements of all three sets, that is, $2^{|G \times R \times A|}$ – this is increased by the number of actions agents have to comply with norms (different actions may have different but overlapping post-conditions and may thus be used interchangeably) and the different ways in which collective actions can be achieved, as defined by domain axioms.

7. ILLUSTRATIVE SCENARIO: GROUP NORMS FOR CLOUD COMPUTING

We illustrate our approach via a cloud computing scenario in which software agents manage Web services to provide support to scientific workflows [Juve and Deelman 2010]. Cloud computing solutions should address three important features, namely, *openness* (that is, components come and go and there is no central control or ownership of all components), *heterogeneity* (that is, components can be implemented using

disparate technologies) and *autonomy* (that is, components decide by themselves on if/when and how they respond or interact with one another).

Ours is a simplified scenario in which we consider three kinds of functionalities, which should come together to provide a solution for a scientific experiment:

- Data storage – agents manage and provide space for storing (intermediate) data results for periods of time; depending on the amount of storage required, many services may need to be combined and the data are split among storage services.
- Data processing – scientific workflows make use of distinct algorithms (implemented as Web services) which are combined together.
- Data provision – data collections are provided for experiments.

Software agents are responsible for providing cloud-based solutions, communicating and coordinating with other agents to deliver functionalities. We endow our agents with organisation- and norm-awareness, enabling them to agree among themselves on individual and collective norm-compliant behaviours.

As mentioned in Section 5, without loss of generality, we model in our scenario norms addressing properties of states, rather than actions.

7.1. Organisation Specification

In our scenario we modelled the following roles:

- Scientific advisor (sa) – sets out the scientific experiment.
- Computing officer (co) – makes technical provisions for the experiment.
- Domain expert (de) – brings the expert’s perspective to the experiment.
- Storage provider (sp) – provides storage.
- Process provider (pp) – provides processing functionalities.
- Data provider (dp) – provides data collections.

These roles are organised along the power structure shown in Figure 2 (where $r \xrightarrow{\preceq} r'$ means $r \preceq r'$).

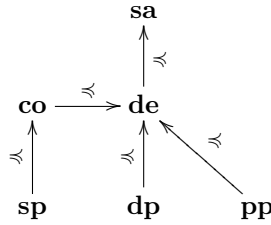


Fig. 2. Power relation among roles in cloud computing scenario

The actions available to our roles are shown in Table III. For simplicity, and because they are not explored in our scenario, the pre-conditions $S_i, 1 \leq i \leq 10$, and the actual labels of actions $ac_i, 1 \leq i \leq 10$, are not shown – these, however, are important when agents need to reason about which actions to choose. We explore group phenomena, whereby agents coordinate to abide by norms.

We inter-relate the post-conditions of the actions above, capturing their *combined effects* (in a similar way as the “count-as” relation [Grossi et al. 2005]). We use logical implications to capture domain axioms, as explained previously. We model in our scenario how cloud solutions are assembled together, that is, how data collections, algorithms and data storage provisions should be combined. We show in Table IV our domain axioms and their intended meaning.

Table III. Actions available in the Cloud Computing Scenario.

Action	Meaning
$\langle S_1, ac_1, \{s_1\} \rangle$	Provision of data storage (type 1)
$\langle S_2, ac_2, \{s_2\} \rangle$	Provision of data storage (type 2)
$\langle S_3, ac_3, \{s_3\} \rangle$	Provision of data storage (type 3)
$\langle S_4, ac_4, \{g_1\} \rangle$	Provision of algorithm 1
$\langle S_5, ac_5, \{g_2\} \rangle$	Provision of algorithm 2
$\langle S_6, ac_6, \{d_1\} \rangle$	Provision of data collection 1
$\langle S_7, ac_7, \{d_2\} \rangle$	Provision of data collection 2
$\langle S_8, ac_8, \{d_3\} \rangle$	Provision of data collection 3
$\langle S_9, ac_9, \{sd_1\} \rangle$	Provision of data collection & storage (solution 1)
$\langle S_{10}, ac_{10}, \{sd_2\} \rangle$	Provision of data collection & storage (solution 2)
$\langle S_{11}, ac_{11}, \{dg_1\} \rangle$	Provision of data collection & algorithm (solution 1)
$\langle S_{12}, ac_{12}, \{dg_2\} \rangle$	Provision of data collection & algorithm (solution 2)
$\langle S_{13}, ac_{13}, \{e_1\} \rangle$	Provision of experiment 1
$\langle S_{14}, ac_{14}, \{e_2\} \rangle$	Provision of experiment 2

Table IV. Domain axioms for solutions in the Cloud Computing Scenario.

Domain Axiom	Meaning
$(s_2 \wedge s_3 \wedge d_3) \rightarrow sd_1$ (DA1)	data collection 3 requires storage provisions of type 2 and 3
$(s_1 \wedge d_1 \wedge d_2) \rightarrow sd_2$ (DA2)	data collections 1 and 2 require storage provisions of type 1
$(d_3 \wedge g_1) \rightarrow dg_1$ (DA3)	data collection 3 requires algorithm 1
$(d_1 \wedge d_2 \wedge g_2) \rightarrow dg_2$ (DA4)	data collections 1 and 2 require algorithm 2
$(sd_1 \wedge dg_1) \rightarrow e_1$ (DA5)	experiment e_1 is performed when sd_1, dg_1 hold
$(sd_2 \wedge dg_2) \rightarrow e_2$ (DA6)	experiment e_2 is performed when sd_2, dg_2 hold

We note that DA5, when used in conjunction with DA1 and DA3, yields $(s_2 \wedge s_3 \wedge d_3 \wedge g_1) \rightarrow e_1$. Such effect will be achieved when actions ac_2, ac_3, ac_4 and ac_8 have been performed (this being captured by their post-conditions). Similarly, DA6, in conjunction with DA2 and DA4, yields $(s_1 \wedge d_1 \wedge d_2 \wedge g_2) \rightarrow e_2$, that is, experiment e_2 is deemed to have been performed when $\{s_1, d_1, d_2\}$ (from sd_2) and $\{d_1, d_2, g_2\}$ (from dg_2) hold; this will be the case when actions ac_1, ac_5, ac_6, ac_7 are performed.

The role labels and actions above are formally related as the set of roles defining the organisational model of the cloud computing scenario, namely:

- Scientific advisor: $\langle sa, \{ac_{13}, ac_{14}\} \rangle$
- Computing officer: $\langle co, \{ac_9, ac_{10}\} \rangle$
- Domain expert: $\langle de, \{ac_{11}, ac_{12}\} \rangle$
- Storage provider: $\langle sp, \{ac_1, ac_2, ac_3\} \rangle$
- Process provider: $\langle pp, \{ac_4, ac_5\} \rangle$
- Data provider: $\langle dp, \{ac_6, ac_7, ac_8\} \rangle$

Figure 3 shows a diagram with organisation components and those agents enacting specific roles. Each node of our diagram is a triple $\langle rl, Agents', S \rangle$, where $rl \in Rl$ is a role, $Agents' \subseteq Agents$ are the agents adopting role rl and $S = cap(\alpha, rl), \alpha \in Agents'$.

We put our scenario specification to use in the following subsection.

7.2. Group Norm Reasoning for Cloud Computing

Algorithms 1–4 rely on an organisation specification as well as information about specific agents enacting the organisation. We make use of the organisation specification introduced in the previous section as well as the following specific agents, and their associated roles:

- $Agents = \{ag_1, \dots, ag_6\}$, where all agents choose norm-compliant behaviours.
- $AssocRl = \{ \langle ag_1, \{sa\} \rangle, \langle ag_2, \{co\} \rangle, \langle ag_3, \{de\} \rangle, \langle ag_4, \{sp, pp\} \rangle, \langle ag_5, \{dp\} \rangle, \langle ag_6, \{pp\} \rangle \}$

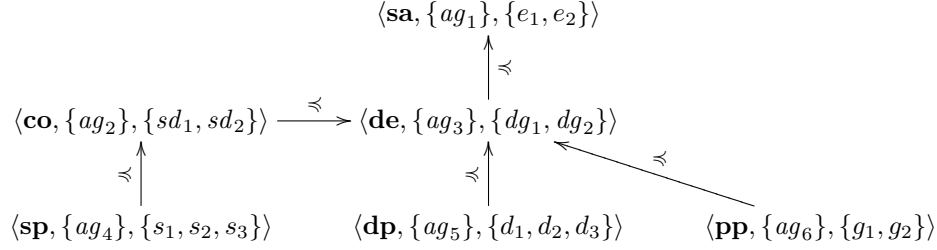


Fig. 3. Organisation and agents of cloud computing scenario

Let us consider a group norm ${}^A\mathbf{O}_G^R e_1$ where:

- The set of agents responsible for the norm is $R = \{\alpha : \text{role}(\alpha, \text{co})\} \cup \{\alpha : \text{role}(\alpha, \text{de})\}$, that is, those agents adopting the roles of computing officer or domain expert.
- The set of addressee agents is $G = \{\alpha : \text{role}(\alpha, \text{sa})\}$, that is, those adopting the role of scientific advisor.
- The set of actors is $A = \{\alpha : \text{role}(\alpha, \text{sp})\} \cup \{\alpha : \text{role}(\alpha, \text{pp})\} \cup \{\alpha : \text{role}(\alpha, \text{dp})\}$, that is, the agents adopting the roles of storage, process or data providers.

This group norm expresses that ‘*the scientific advisors must ensure the provision of experiment 1, under responsibility of the computing officers and domain experts*’. The actorship relation, left implicit in the norm, follows from the organisational structure as presented in Figure 2.

For the set definitions above we make use of property $\text{role}(\alpha, \text{rl})$ which holds iff $\langle \alpha, \text{rl} \rangle \in \text{AssocRl} \wedge \text{rl} \in \text{Rl}$. Given the set *Agents*, we have:

- The agents responsible are $\text{value}(R, \text{Agents}) = \{ag_2, ag_3\}$
- The set of addressee agents is $\text{value}(G, \text{Agents}) = \{ag_1\}$
- The actor agents are $\text{value}(A, \text{Agents}) = \{ag_4, ag_5, ag_6\}$

We explore what happens when agent ag_1 (the scientific advisor) invokes Alg. 1 to reason about the norm. Since ag_1 belongs to the set G of addressee agents, Alg. 4 is invoked. Step 1 of Alg. 4 computes a subset of those agents in R over which ag_1 has power, that is, $\text{ResponsibleSet} = \{ag_3\}$ (since ag_3 has role de and $\text{de} \preceq \text{sa}$). Alg. 4 then invokes (in line 3) Alg. 3 via $\text{responsibility}(ag_3, {}^A\mathbf{O}_G^R e_1)$.

Line 1 of Alg. 3 has ag_3 attempting to coordinate with $R = \{ag_2\}$ about who is to be in charge over which norms. Let us assume that agents ag_2 and ag_3 work out (via the domain axioms) that to achieve e_1 they ought to achieve $(sd_1 \wedge dg_1)$, that is, the returned norm ${}^{A'}\mathbf{X}_{G'}^{R'} \phi$ (we use \mathbf{X} to differentiate the modality of the norm from the modality of the input parameter norm \mathbf{D}) would be ${}^A\mathbf{O}_G^R (sd_1 \wedge dg_1)$; the choice of which agent to be in charge is dependent on who has power over which other agents. In this instance, if the agents had agreed on ag_2 being in charge, then there would not have been a way of fulfilling the norm as ag_2 would not have power over agents with the required capabilities. In our scenario, the agents agree that ag_3 is to be in charge.

Line 3 of Alg. 3 computes the set of actors over which ag_3 has power, that is, $\{ag_4, ag_5, ag_6\}$ (where ag_4 comes in via ag_2 as power is transitive – cf. Def. 5.4). A loop ensues, whereby the actorship algorithm (Alg. 2) is invoked with each agent in $\{ag_4, ag_5, ag_6\}$.

We now follow what happens when the actorship algorithm (Alg. 2) is invoked by ag_4 . Line 1 computes all minimal coalitions of agents associated with roles whose capabilities achieve $(sd_1 \wedge dg_1)$. The domain axioms help agents to further explore the different ways in which post-conditions (that is, the effects) of actions can be achieved – in our

scenario we have $\{s_2, s_3, d_3, g_1\} \vdash (sd_1 \wedge dg_1)$ and the set of coalitions is $\{\{ag_4, ag_5, ag_6\}\}$. Agent ag_4 would compute its own capabilities (line 4) that is, $MyCap = \{s_1, s_2, s_3\}$, and would try to coordinate with the other agents of the coalition to agree on how each could contribute to achieving $(sd_1 \wedge dg_1)$, that is, $coordinate(ag_4, \{ag_4, ag_5, ag_6\}, contribute, \{s_1, s_2, s_3\}, (sd_1 \wedge dg_1))$.

We show in Figure 4 how the various reasoning steps were carried out in our scenario, also including which agent/role led the reasoning step and who was also involved. The numbers on the arrows indicate the order in which the various agents got

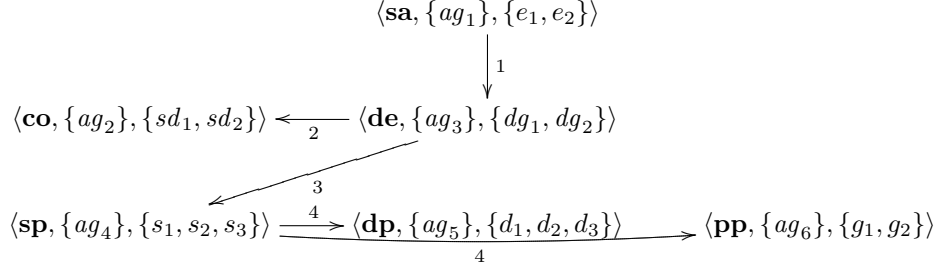


Fig. 4. Flow of reasoning in cloud computing scenario

involved in the normative reasoning, also indicating who (that is, the agent who is in the origin of the arrow) called on for help from whom (the agent who the tip of the arrow points at). We note that arrow 1 indicates addressment reasoning, lines 2 and 3 indicate responsibility reasoning and lines 4 indicate actorship reasoning. We note that the agent associated with role *de* was able to take advantage of the transitivity of the power relation, roping in an actor agent enacting role *sp* (via role *co*, over which *de* has power).

Using our same scenario with norms $^A\mathbf{O}_G^R(e_2 \vee sd_1)$ and $^A\mathbf{F}_G^R s_1$ (where groups A, R and G are the same as before), and following a similar reasoning process, we would see in steps 3 and 4 (cf. the diagram in Fig. 4) an interesting phenomenon. In step 3 two potential coalitions, namely $\{ag_4, ag_5, ag_6\}$ (with combined capabilities to provide for e_2) and $\{ag_4, ag_5\}$ (with combined capabilities to provide for sd_1) would be computed (line 1 of Alg. 2). However, when acting agents factor in the prohibition via a coordination process (as explained in Section 5.3), then they will work out that ag_4 should refrain from achieving s_1 , hence coalition $\{ag_4, ag_5, ag_6\}$ would not achieve e_2 . Coalition $\{ag_4, ag_5\}$ would then coordinate to provide for sd_1 , thus fulfilling the obligation, and abiding by the prohibition. The coordination process invoked in the reasoning mechanisms, as explained in Section 5.3, support this activity.

8. RELATED WORK

In addition to the various pieces of research discussed throughout the paper, in this section we address other related work. Work on collective agency (e.g., [Carmo 2010; Carmo and Pacheco 2001; Pacheco and Carmo 2003]) and collective obligations (e.g., [Grossi et al. 2004]) have addressed similar concerns as ours. These approaches represent norms over actions, establishing groups of agents to whom the norms apply. Some approaches regard group norms as a shorthand for a norm which applies to all/some members of the group (e.g., [Carmo and Pacheco 2001]), whereas other approaches (e.g., [Grossi et al. 2004]) regard group norms (more specifically, collective obligations) as a shared complex action requiring individual contributions (i.e., simpler actions) from

those individuals of the group. However, these approaches only deal with the element of shared responsibility, neglecting the element of shared actorship.

Research about the concept of shared actorship can be found in work on joint action and coalitions (e.g., [Borgo 2007; Ågotnes and Alechina 2011; Grossi et al. 2007]). This line of investigation is relevant as it looks into individual deliberation when coordination is required. Work exploring aspects of delegation (e.g., [Dignum and Dignum 2011; Norman and Reed 2010]) sheds light on how norms can be transferred among individuals and groups. When agents join organisations they will need to consider the implications of taking up roles, since these will determine to which groups agents will ultimately belong, and consequently which norms will be applicable, as well as how power and delegation will impact on the agents' choices. Research has addressed issues of expressiveness and reasoning complexity in various logics of coalition (e.g., [Broersen et al. 2009; Troquard 2014]), establishing that even for simple propositional fragments, complexity is very high (i.e., PSPACE in the size of the formula checked).

The notion of group association and imposing norms on groups of agents is closely related to the concept of *roles*. Roles have been explored in research on electronic institutions [Esteva et al. 2001] and organisations [Dignum 2004; Hannoun et al. 2000; McCallum et al. 2008; Pacheco and Carmo 2003]. Roles describe collections of *stereotypical* individuals who, by adopting a role, become subject to any norms associated with that role. We note that norms addressing roles are a useful shorthand for specialised norms addressing individuals, that is, they stand for “any one who has adopted role r is subject to norm ν ”. For instance, a norm such as “Soldiers are forbidden to enter area (x, y) ” and given agents a_1, \dots, a_n who have taken up the *soldier* role, stands for “Agent a_i is forbidden to enter area (x, y) ”, for each $i, 1 \leq i \leq n$. Importantly, in existing research role norms typically do not influence the joint behaviour of individuals and do not require coordination.

In philosophy of mind, a recurrent way of looking at groups of agents is “group agency”, whereby the group of agents is considered as an agent itself. [List and Pettit 2011] theorize, however, that true group agency cannot exist because of the individuality of the members of the group; in order to achieve the required commonality to see a group as an agent, the group needs a level of collective rationality, anonymity and systematicity that cannot be achieved (under reasonable assumptions). A group cannot be simply reduced to its members (there is more than just a “mere collection”) because of coordination activities required to make the group function properly. In this paper we explored exactly those aspects of groups that make them more than “mere collection” of agents, by trying to formalise the rationality required to function as a group.

A similar vision of groups is taken by the field of commitment-based norms [Singh et al. 2009; Noriega et al. 2013; Baldoni et al. 2014; Chopra et al. 2014]. In the formalisation of norms with commitments, the authors often use individuals and companies interchangeably, for instance to express that an insurance company is committed to cover the expenses of the claim of a customer. It appears that these formalisations use groups as a complex entity that can be held accountable (as a group) for the actions (of the group). None of the papers, however, go into the detail as to how the action or responsibility is propagated to the individual members of the group, as is done here.

An interesting field of related research is on organizational-aware agency [van Riemsdijk et al. 2009; Jensen et al. 2014]. The research develops frameworks to build agents that are aware of organizational restrictions and norms to allow them to reason about how to be compliant with an organizational specification. Since they work with existing organization specification languages, e.g., OperA [Dignum 2004] or MOISE [Hannoun et al. 2000], and these specification languages use role norms as individual norms (agents playing that role are each addressed individually, are assumed to

act individually and are held responsible individually). More interesting are the co-ordination restrictions put on the agents by other components of the organization – our contributions to reasoning about addressment, responsibility or actorship in group norms enrich organization-aware agency frameworks.

9. CONCLUSIONS AND FUTURE WORK

We have proposed a representation for group norms, a topic largely ignored in the literature. Our proposal caters for three distinct types of stakeholders, namely, the addressees of the norm, those responsible for the norm, and those whose behaviours are impacted by the norm (the actors). We used a simple set-based formalism to precisely define groups and explored our notion of group norms within a minimalist organisation model. Using common organisational concepts such as roles, capabilities and power, we proposed three inter-connected simple reasoning mechanisms for autonomous agents to operate within organisations factoring group norms when deciding who is to do (or not do) what, with whom, and when.

Our representation has been influenced by a taxonomy of cases for group norms [Aldewereld et al. 2013], with two dimensions – the individual and the collective – within a group. Certain norms, although addressed to groups, are fulfilled/violated by a single (or some) members; other norms are aimed at the group as a whole. Our reasoning mechanisms are a first attempt at defining how agents can factor in group/individual issues when deciding what to do within an organisation (hence there is a degree of predictability on the agents' part), presenting clear connections with generally agreed organisational concepts. Very importantly, our approach can be adapted and applied to other circumstances where group norms are needed, for instance, by dropping one or more of the groups in the formalism, with appropriate simplifications in the reasoning mechanisms. The reasoning mechanisms can be applied to other situations where group norms are used, but our formalisation makes it possible to reason with and about the various groups (addressees, actors, and those responsible for the norm) also allowing agents to differentiate between these.

In this paper, we have abstracted from what the action or state referred to by the norm entails. Depending on the interpretation (cf. [Aldewereld et al. 2013]) the action may be individual, *i.e.*, each agent in the Actorship group should act on it, or collective, in which case all, or a subset, of the agent in the Actorship group should jointly act on it. In future work, we will further detail the reasoning about group norms such that planning for the norm can incorporate both reasoning about individual plans and joint plans.

Our representation of norms may give rise to *conflicts*, whereby actions (or states of affairs) are simultaneously obliged and prohibited (or permitted and prohibited) [Vasconcelos et al. 2009]. We note that in our norm representation, for two norms ${}^A\mathbf{O}_G^R \varphi$ and ${}^{A'}\mathbf{F}_{G'}^{R'} \phi$ to be in conflict, the following should hold (where Φ is the background theory): *i)* $\Phi \wedge \varphi \wedge \phi \vdash \perp$, that is, the norms' actions or states of affairs, combined with the background theory, derive an inconsistency \perp ; *ii)* $(\text{value}(A, \text{Agents}) \cap \text{value}(A', \text{Agents})) \neq \emptyset$, that is, the sets of actors of both norms must have an overlap. In future work we shall explore alternative mechanisms to manipulate the group descriptions, so as to remove any overlap. Moreover, we will explore means whereby groups R and G might agree among themselves who will handle specific norms, thus partitioning the space of possible coalitions for acting agents, and thus avoiding conflicts altogether.

We assumed our agents “do as they are told” when asked by an agent who has power to do so, power being represented as a relation among roles of an organisation. Our research will relax this assumption, addressing situations in which agents may fail

to carry out individual actions required for a group obligation, or when agents do not refrain from performing individual actions which together violate a group prohibition. In such situations, sanctions should be *apportioned*, that is, divided among those in the various group, following different rationales [Dubnick and Frederickson 2014]. If we consider just the group of actors, some cases are clear: if none of the members of a group carried out their individual actions in order to comply with a group obligation, then they should all be equally sanctioned. Other cases are not as obvious: given a group prohibition on a collective action, it is not clear if only the last agent to carry out an individual action, which together with other individual actions of other members of the group violated a prohibition, should be sanctioned, or if the whole group should be sanctioned. Sanctions should be apportioned to those agents addressed by a norm if they failed to procure agents responsible for the norm, and similarly, agents responsible for a norm should have sanctions apportioned to their group if they fail to procure actors.

We are currently extending our mechanisms with communication, using classic, off-the-shelf protocols such as the Contract Net Protocol (CNP¹⁴) when those agents responsible for a norm call on the help of actors (*e.g.* choosing which agent α to invoke *actorship* with, in Algorithm 4), and voting protocols/mechanisms [Battaglini et al. 2005] when those agents responsible decide among themselves who should pursue the norm (the *coordinate* step in Algorithm 3). We will connect our approach with existing planning techniques (*e.g.*, HTN [Nau et al. 2003]) to evaluate how our group norms can help agents agree on joint plans with fewer messages and in fewer rounds.

Acknowledgements: The authors would like to thank the three anonymous reviewers for their comments, suggestions and constructive criticisms. Thanks are due to Dr. Nir Oren, for comments on earlier versions of the paper, and Mr. Seumas Simpson, for proof-reading the manuscript. Any remaining mistakes are the sole responsibility of the authors. W. W. Vasconcelos would like to acknowledge the support of the Engineering and Physical Sciences Research Council (EPSRC-UK) within the research project “Scrutable Autonomous Systems” (grant ref. EP/J012084/1).

REFERENCES

- Thomas Ågotnes and Natasha Alechina. 2011. Reasoning about joint action and coalitional ability in Kn with intersection. In *Procs. 12th Int'l Conf. on Computational Logic in Multi-Agent Systems (CLIMA'11) (Lecture Notes in Computer Science)*, Vol. 6814. Springer-Verlag, Berlin, Heidelberg, 139–156.
- Huib Aldewereld, Virginia Dignum, and Wamberto Weber Vasconcelos. 2013. We Ought To; They Do; Blame the Management! - A Conceptualisation of Group Norms. In *Coordination, Organizations, Institutions, and Norms in Agent Systems IX - COIN 2013 International Workshops, COIN@AAMAS, St. Paul, MN, USA, May 6, 2013, COIN@PRIMA, Dunedin, New Zealand, December 3, 2013, Revised Selected Papers (Lecture Notes in Computer Science)*, Tina Balke, Frank Dignum, M. Birna van Riemsdijk, and Amit K. Chopra (Eds.), Vol. 8386. Springer, 195–210. DOI: http://dx.doi.org/10.1007/978-3-319-07314-9_11
- A.R. Anderson. 1958. A Reduction of Deontic Logic to Alethic Modal Logic. *Mind* 67 (1958), 100–103.
- Matteo Baldoni, Cristina Baroglio, and Federico Capuzzimati. 2014. A Commitment-based Infrastructure for Programming Socio-Technical Systems. *ACM Transactions on Internet Technology (TOIT)* 14, 4 (2014), 23.
- Marco Battaglini, Rebecca Morton, and Thomas Palfrey. 2005. *Efficiency, Equity, and Timing in Voting Mechanisms*. Working Papers 81. Princeton Univ., Dept. of Economics, Center for Economic Policy Studies. <http://ideas.repec.org/p/pri/cepsud/81.html>
- Nuel Belnap and Michael Perloff. 1988. Seeing to it that: a canonical form for agentives. *Theoria* 54, 3 (1988), 175–199.
- G. Boella and L. van der Torre. 2004. Normative multiagent systems. In *Proceedings of Trust in Agent Societies Workshop at AAMAS'04*. New York.

¹⁴<http://www.fipa.org/specs/fipa00029/SC00029H.pdf>

- Stefano Borgo. 2007. Coalitions in action logic. In *Procs. 20th Int'l Joint Conf. on Artificial Intelligence (IJ-CAI'07)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1822–1827. <http://dl.acm.org/citation.cfm?id=1625275.1625570>
- Jan Broersen, Frank Dignum, Virginia Dignum, and John-Jules Ch. Meyer. 2004. Designing a Deontic Logic of Deadlines. In *Deontic Logic in Computer Science (Lecture Notes in Computer Science)*, Alessio Lomuscio and Donald Nute (Eds.), Vol. 3065. Springer Berlin Heidelberg, 43–56. http://dx.doi.org/10.1007/978-3-540-25927-5_5
- Jan Broersen, Andreas Herzig, and Nicolas Troquard. 2009. What groups do, can do, and know they can do: an analysis in normal modal logics. *Journal of Applied Non-Classical Logics* (2009), 261–290.
- Henrique Lopes Cardoso and Eugénio Oliveira. 2009. Flexible Deadlines for Directed Obligations in Agent-based Business Contracts. In *Procs. 8th Int'l Conf. on Autonomous Agents & Multiagent Systems (Vol. 2) (AAMAS '09)*. Int'l Foundation for Autonomous Agents & Multiagent Systems, Richland, SC, 1307–1308. <http://dl.acm.org/citation.cfm?id=1558109.1558266>
- José Carmo. 2010. Collective agency, direct action and dynamic operators. *Logic Journal of the IGPL* 18, 1 (2010), 66–98.
- José Carmo and Olga Pacheco. 2001. Deontic and Action Logics for Organized Collective Agency, Modeled through Institutionalized Agents and Roles. *Fundam. Inform.* 48, 2-3 (2001), 129–163.
- Amit K Chopra, Fabiano Dalpiaz, F Basak Aydemir, Paolo Giorgini, John Mylopoulos, and Munindar P Singh. 2014. Protos: Foundations for engineering innovative sociotechnical systems. In *Requirements Engineering Conference (RE), 2014 IEEE 22nd International*. IEEE, 53–62.
- Keith L. Clark. 1978. Negation as Failure. In *Logic and Data Bases*, Hervé Gallaire and Jack Minker (Eds.). Springer US, 293–322. DOI: http://dx.doi.org/10.1007/978-1-4684-3384-5_11
- Tiago de Lima, Lambér Royakkers, and Frank Dignum. 2010. A logic for reasoning about responsibility. *Logic Journal of IGPL* 18, 1 (2010), 99–117. DOI: <http://dx.doi.org/10.1093/jigpal/jzp073>
- Robert Demolombe and Vincent Louis. 2006. Norms, Institutional Power and Roles: Towards a Logical Framework. In *Foundations of Intelligent Systems*. Lecture Notes in Computer Science, Vol. 4203. Springer, 514–523. DOI: http://dx.doi.org/10.1007/11875604_58
- Virginia Dignum. 2004. *A Model for Organizational Interaction: Based on Agents, Founded in Logic*. Ph.D. Dissertation. Universiteit Utrecht, The Netherlands.
- Virginia Dignum and Frank Dignum. 2011. A Logic of Agent Organizations. *Logic Journal of IGPL* (2011). DOI: <http://dx.doi.org/10.1093/jigpal/jzr041>
- Melvin J. Dubnick and H. G. Frederickson. 2014. *Accountable Governance: Problems and Promises*. Routledge.
- E. Allen Emerson. 1990. Temporal and modal logic. In *Handbook of Theoretical Computer Science*, J. van Leeuwen (Ed.). Vol. B. MIT Press, 955–1072.
- Marc Esteva, Juan A. Rodríguez-Aguilar, Carles Sierra, Pere Garcia, and Josep Lluís Arcos. 2001. On the Formal Specifications of Electronic Institutions. In *Agent Mediated Electronic Commerce (Lecture Notes in Computer Science)*, Vol. 1991. Springer, 126–147. DOI: http://dx.doi.org/10.1007/3-540-44682-6_8
- Noah E. Friedkin. 1986. A Formal Theory of Social Power. *Journal of Mathematical Sociology* 12, 2 (1986), 103–126.
- A. García-Camino, P. Noriega, and J.-A. Rodríguez-Aguilar. 2005. Implementing norms in electronic institutions. In *Procs. 4th Int'l Joint Conf. on Autonomous Agents & Multiagent Systems (AAMAS '05)*. ACM, New York, NY, USA, 667–673. DOI: <http://dx.doi.org/10.1145/1082473.1082575>
- Andrés García-Camino, Juan Antonio Rodríguez-Aguilar, and Wamberto Vasconcelos. 2008. A Distributed Architecture for Norm Management in Multi-Agent Systems. In *Coordination, Organizations, Institutions, and Norms in Agent Systems III*, Jaime Simão Sichman, Julian Padget, Sascha Ossowski, and Pablo Noriega (Eds.). Lecture Notes in Computer Science, Vol. 4870. Springer Berlin Heidelberg, 275–286. DOI: http://dx.doi.org/10.1007/978-3-540-79003-7_20
- Malik Ghallab, Dana Nau, and Paolo Traverso. 2004. *Automated Planning: Theory & Practice*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Davide Grossi, Frank Dignum, Lambèr Royakkers, and Jean-Jules Meyer. 2004. Collective obligations and agents: Who gets the blame?. In *Deontic Logic in Computer Science (Lecture Notes in Computer Science)*, Vol. 3065. Springer, 129–145. DOI: http://dx.doi.org/10.1007/978-3-540-25927-5_9
- Davide Grossi and Andrew Jones. 2013. *Constitutive Norms and Counts-as Conditionals*. Vol. 1. College Publications, London, 407–441.
- Davide Grossi, John-Jules Ch. Meyer, and Frank Dignum. 2005. Modal Logic Investigations in the Semantics of Counts-as. In *Procs 10th Int'l Conf. on AI & Law (ICAIL)*. ACM, New York, NY, USA, 1–9. DOI: <http://dx.doi.org/10.1145/1165485.1165487>

- Davide Grossi, Lambér Royakkers, and Frank Dignum. 2007. Organizational structure and responsibility. *Artificial Intelligence and Law* 15, 3 (2007), 223–249. DOI: <http://dx.doi.org/10.1007/s10506-007-9054-0>
- Paul Halmos. 1960. *Naïve Set Theory*. Van Nostrand. Reprinted by Springer-Verlag, Undergraduate Texts in Mathematics, 1974.
- Xu Han, S. Mandal, K. R. Pattipati, D. L. Kleinman, and M. Mishra. 2014. An Optimization-Based Distributed Planning Algorithm: A Blackboard-Based Collaborative Framework. *IEEE Transactions on Systems, Man, and Cybernetics* 44, 6 (June 2014), 673–686. DOI: <http://dx.doi.org/10.1109/TSMC.2013.2276392>
- Mahdi Hannoun, Olivier Boissier, Jaime Simão Sichman, and Claudette Sayettat. 2000. MOISE: An Organizational Model for Multi-agent Systems. In *IBERAMIA-SBIA (LNCS)*, Vol. 1952. Springer.
- John F. Harty. 2001. *Agency and Deontic Logic*. Oxford University Press, Oxford.
- Andreas Schmidt Jensen, Virginia Dignum, and Jørgen Villadsen. 2014. The AORTA architecture: Integrating organizational reasoning in Jason. In *Engineering Multi-Agent Systems*. Springer, 127–145.
- Andrew J. I. Jones and Marek J. Sergot. 1996. A Formal Characterisation of Institutionalised Power. *Logic Journal of the IGPL* 4, 3 (1996), 427–443.
- Gideon Juve and Ewa Deelman. 2010. Scientific Workflows and Clouds. *Crossroads* 16, 3 (March 2010), 14–18. DOI: <http://dx.doi.org/10.1145/1734160.1734166>
- Martin Kollingbaum and Tim Norman. 2003. NoA – A Normative Agent Architecture. In *Procs. 18th Int'l Joint Conf on Artificial Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1465–1466. <http://dl.acm.org/citation.cfm?id=1630659.1630899>
- Christian List and Philip Pettit. 2011. *Group agency: The possibility, design, and status of corporate agents*. Oxford University Press Oxford.
- Alessio Lomuscio and Marek Sergot. 2002. On Multi-agent Systems Specification via Deontic Logic. In *Intelligent Agents VIII*. Lecture Notes in Computer Science, Vol. 2333. Springer, 86–99. DOI: http://dx.doi.org/10.1007/3-540-45448-9_7
- Fabiola López y López. 2003. *Social Power and Norms: Impact on Agent Behaviour*. Ph.D. Dissertation. University of Southampton, UK.
- Mairi McCallum, Wamberto Weber Vasconcelos, and Timothy J. Norman. 2008. Organizational change through influence. *Autonomous Agents and Multi-Agent Systems* 17, 2 (2008), 157–189.
- Paul McNamara. 2006. Deontic logic. In *Logic and the Modalities in the Twentieth Century*, Dov M. Gabbay and John Woods (Eds.). Handbook of the History of Logic, Vol. 7. North-Holland, 197 – 288. DOI: [http://dx.doi.org/10.1016/S1874-5857\(06\)80029-4](http://dx.doi.org/10.1016/S1874-5857(06)80029-4)
- Felipe Meneguzzi, Odinaldo Rodrigues, Nir Oren, Wamberto W. Vasconcelos, and Michael Luck. 2015. BDI reasoning with normative considerations. *Engineering Applications of Artificial Intelligence* 43, 0 (2015), 127–146. DOI: <http://dx.doi.org/10.1016/j.engappai.2015.04.011>
- Dana Nau, Oktay Ilghami, Ugur Kuter, J. William Murdock, Dan Wu, and Fusun Yaman. 2003. SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research* 20 (2003), 379–404.
- Pablo Noriega, Amit K Chopra, Nicoletta Fornara, Henrique Lopes Cardoso, and Munindar P Singh. 2013. Regulated MAS: Social Perspective. *Normative multi-agent systems* 4 (2013), 93–133.
- Timothy J. Norman and Chris Reed. 2010. A logic of delegation. *Artificial Intelligence* 174 (Jan. 2010), 51–71. Issue 1. DOI: <http://dx.doi.org/10.1016/j.artint.2009.10.001>
- Nir Oren, Michael Luck, and Simon Miles. 2010. A model of normative power. In *Procs. 9th Int'l Conf on Autonomous Agents & Multiagent Systems (AAMAS '10)*. IFAAMAS, Richland, SC, 815–822. <http://dl.acm.org/citation.cfm?id=1838206.1838315>
- E. Ostrom. 2005. *Understanding institutional diversity*. Princeton Univ Pr.
- Olga Pacheco and José Carmo. 2003. A Role Based Model for the Normative Specification of Organized Collective Agency and Agents Interaction. *Autonomous Agents and Multi-Agent Systems* 6 (March 2003), 145–184. Issue 2. DOI: <http://dx.doi.org/10.1023/A:1021884118023>
- Munindar P Singh, Amit K Chopra, and Nirmal Desai. 2009. Commitment-Based Service-Oriented Architecture. *IEEE Computer* 42, 11 (2009), 72–79.
- Paolo Torroni, Federico Chesani, P. Yolum, Marco Gavanelli, Munindar P. Singh, Evelina Lamma, M. Alberti, and P. Mello. 2009. *Modelling Interactions via Commitments and Expectations*. IGI Global.
- Nicolas Troquard. 2014. Reasoning about coalitional agency and ability in the logics of “bringing-it-about”. *Autonomous Agents and Multi-Agent Systems* 28, 3 (2014), 381–407. DOI: <http://dx.doi.org/10.1007/s10458-013-9229-x>
- M. Birna van Riemsdijk, Koen Hindriks, and Catholijn Jonker. 2009. Programming organization-aware agents. In *Engineering Societies in the Agents World X*, Huib Aldewereld, Virginia Dignum, and Gauthier Picard (Eds.). Springer, 98–112.

- Wamberto W. Vasconcelos, Andrés García-Camino, Dorian Gaertner, Juan A. Rodríguez-Aguilar, and Pablo Noriega. 2012. Distributed Norm Management for Multi-agent Systems. *Expert Syst. Appl.* 39, 5 (April 2012), 5990–5999. DOI: <http://dx.doi.org/10.1016/j.eswa.2011.11.108>
- Wamberto W. Vasconcelos, Martin J. Kollingbaum, and Timothy J. Norman. 2009. Normative conflict resolution in multi-agent systems. *Autonomous Agents and Multi-Agent Systems* 19, 2 (2009), 124–152. DOI: <http://dx.doi.org/10.1007/s10458-008-9070-9>
- Ryan K. Williams. 2014. *Interaction and Topology in Distributed Multi-Agent Coordination*. Ph.D. Dissertation. Department of Electrical Engineering, University of Southern California.