An Algorithm for Conflict Resolution in Regulated Compound Activities

Andrés García-Camino, Pablo Noriega, and Juan-Antonio Rodríguez-Aguilar

IIIA-CSIC, Campus UAB, 08193 Bellaterra, Spain
`{andres, pablo, jar}@iiia.csic.es`

Abstract. The use of norms is a well-known technique of co-ordination in multi-agent systems (MAS) adopted from human societies. A normative position is the "social burden" associated with individual agents, that is, their obligations, permissions and prohibitions. Compound activities may be regulated by means of normative positions. However, conflicts may appear among normative positions of activities and sub-activities. Recently several computational approaches have appeared to make norms operational in MAS but they do not cope with compound activities. In this paper, we propose an algorithm to determine the set of applicable normative positions, i.e., the largest set of normative positions without conflicts in the state of an activity, and propagate them to the sub-activities.

1 INTRODUCTION

Society has frequently come across the need of coordinating interactions among individuals and one way of addressing that need has been to establish restrictive environments where the interactions are constrained to only those participants and those interactions that are meant to be. For analogous reasons, the MAS community has proposed regulated environments where agents –human or software– interact as [1,2,3].

The environments we will have in mind in this paper are regulated environments where agent interactions are structured as repetitive interactions –that we shall call *activities*– and the whole environment is the result of the composition of many such activities. These activities are subject to explicit sets of conventions that prescribe how the actions of agents that participate in a given activity establish or fulfil commitments that affect the participants of that activity and of subordinate activities. For lack of a better term we will refer to such environments as regulated compound activities.

Many real world societies conform to this type of regulated environments and virtual counterparts are easy to conceive. For instance, Figure 1 describes the example of an on-line commodities trading market that has different price-fixing conventions which may have different simultaneous enactments (different auctions to buy, say, wholesale fruit and poultry; one-to-many negotiations for supermarkets to stock their weekly supply, direct purchasing for scarce quality

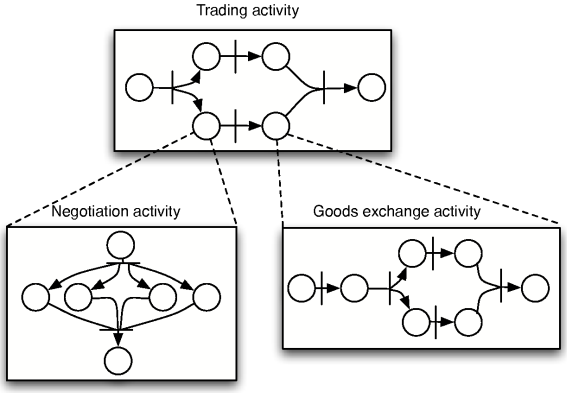


Fig. 1. Example of compound activities

goods like spices). These price-fixing activities are, on one hand, preceded by activities whose purpose is to set the grounds for the day's trading (e.g., activities to introduce whatever is to be exchanged during the day, or the accreditation of buyers and their credit lines, etc.) and, on the other hand, they may be followed by other activities like delivery contracting, temporary warehousing or packaging, which in-turn may be compound activities on their own, etc. Other examples of compound activities that naturally come to mind are hospital operation, the football world association (FIFA) activities, or the execution of everyday local government activities.

The conventions that regulate activities, as the examples show, usually have both a procedural component and a declarative one. The conventions may be expressed in different ways although the most familiar ones are commitment-based interaction protocols (e.g., [1,4,3]) and logical (and logic-based) systems (e.g., [5,6]), or as a combination of both [7]. Some of these approaches have elegant conceptual frameworks behind and a few have also an operationalisation that is amenable to be implemented and still a few have been able to integrate the three previous types of convention representation. This last family is what we aim at in our proposal.

In advancing conceptual or implementation frameworks for compound activities, one of the main problems to address —from a social perspective— is to keep track of the commitments that are being established and fulfilled dynamically anywhere in the (compound) society while the society is active. This is a particularly significant problem in societies where truly autonomous entities intervene. The actual problem, then, is to keep an appropriate record of the commitments that are being made and their follow-up, to make sure that the commitments are consistent. This entails the need to make the problem operational, state it in such a way that formal and implementations are feasible and practical.

In this paper, consequently, we want to make headway towards a proposal of a framework for commitment management in regulated environments formed by compound activities. For this purpose we adopt a social perspective to the problem and take a simplified and unconventional approach along the following lines: We abstract the notion of commitment and commitment management by focusing only on the prohibitions, permissions and obligations associated to actions, what we will call the *normative positions*. We also abstract the way normative positions propagate in the society by having a directed graph linking the activities that inherit normative positions and assuming that the graph is *predefined* and to that extent independent of the way activities are actually connected in formal and implementational ways. The management of commitments is also abstracted in this paper by focusing only in the resolution of conflicts among normative positions and using conventional priority criteria to choose between conflicting positions and only then propagate normative positions to the subordinate activities. Since we are interested in making our proposal operational we also present an algorithm that implements these ideas and keeps track of the evolution of normative positions in acyclic compound activities and maintains a conflict-free normative positions base.

The rest of the paper is structured as follows. In section 2, normative positions, deontic conflicts and criteria for conflict resolution are introduced. Compound activities and their deontic conflicts are defined in section 3. In this section, properties of normative consistency in regulated compound activities, as e.g. strong and weak conflict-freedom, are also introduced. In section 4, an algorithm to resolve deontic conflicts in compound activities is proposed. In section 5 we present an example of how the algorithm works. Finally, conclusions and future work are outlined in the last section.

2 NORMATIVE POSITIONS

A normative position is the “social burden” associated with individual agents, that is, their obligations, permissions and prohibitions (cf. [5]). Depending on what agents do, their normative positions may change – for instance, permissions/prohibitions can be revoked or obligations, once fulfilled, may be removed.

In regulated actions, the change of normative positions maybe determined by rules that, for example, are time-dependent. For instance, a permission to lend a book is enabled every week day at 9:00a.m. and disabled every week day at 9:00p.m. Action performances can enable normative positions that can be subsequently fulfilled or cancelled. For example, an obligation to pay for a good is enabled if that good is received (generation). This obligation can be disabled either by paying for the good (fulfilment) or by returning it (cancellation).

Deontic conflicts may appear when norms enable new normative positions that are incompatible with the normative positions already enabled. Traditionally, three principles have been used to resolve deontic conflicts: *legis posterior*, *legis specialis* and *legis superior*. These principles order the norms to avoid conflicts following three criteria: a *chronological* criterion (*lex posterior*), a *speciality* criterion (*lex specialis*) by which a specific law prevails over a general law and a *source* criterion, where preference is linked to the rank of the issuing authority (*lex superior*). We extend these criteria with an extra criterion, the *salience* criterion where we may capture whatever other notions of pertinence or relevance may be of use in a specific activity.

Although such criteria are used to resolve deontic conflicts, in some occasions two or more criteria may need to be sequentially applied to achieve that goal. In those circumstances the criteria involved need to be totally ordered. An example of this meta-ordering is when the source criterion prevails over the speciality, salience and the chronological ones, the salience criterion prevails over the speciality and the chronological ones, and the speciality criterion prevails over the chronological one.

In this paper, the chronological and salience criteria will have a straightforward operationalisation. The speciality criterion will correspond to the hierarchical dependence of an activity and its subactivities. Establishing certain agents as sources of law, defining the ordering of sources of law, and ordering norms using the source criterion are left for future work.

We may now illustrate these ideas and state them in a more precise way. We mentioned that a normative position is a permission, a prohibition or an obligation to perform a specific action. Since we are concerned with resolving conflicts between normative positions, we find useful to associate to every normative position a time stamp that corresponds to the moment it becomes effective (enabled). For the same reason we find useful to associate to normative positions an argument that stands for its salience, although it is beyond the scope of this paper how values may be assigned to that parameter. More precisely:

Definition 1. *Let* $\delta \in \{per, prh, obl\}$ *be a label for the "social burden" of performing an action identified by* a*, a salience constant* $s \in \mathbb{N}$ *and a time-stamp* $t \in \mathbb{N}$*, the formula* $\delta(a, s, t)$ *stands for a normative position that states that at time* t*, and with priority* s*, action* a *becomes permitted, obligatory or prohibited.*

Examples of normative positions may be: $per(bid_{ag_1}, 0, 0)$, $prh(bid_{ag_1}, 2, 1)$, etc. The former normative position intuitively states that agent ag_1 is permitted to bid since time 0 and this normative position has priority 0. The latter normative position intuitively means that agent ag_1 is prohibited to bid since time 1 and this normative position has priority 2.

As mentioned above, deontic conflicts among normative positions can arise as agent interactions progress. We will say that two normative positions are in conflict if one is a permission or obligation and the other is a prohibition over the same action than the former, regardless of their corresponding salience and enabling times. That is:

Definition 2. *Given two normative positions* np, np' *such that* $np = \delta(a, s, t)$ *and* $np' = \delta'(a', s', t')$ *;* np, np' *are in conflict, denoted* $np \ggg\!\lll np'$*, iff:*

1. $\{\delta\} \cup \{\delta'\} = \{per, prh\}$, $a = a'$; *or*
2. $\{\delta\} \cup \{\delta'\} = \{obl, prh\}$, $a = a'$.

In the previous example, $per(bid_{ag_1}, 0, 0)$ and $prh(bid_{ag_1}, 2, 1)$ are in conflict by the first condition of definition 2.

We take care of the other two parameters of normative positions with the following definitions.

Definition 3. *A normative position* $np = \delta(e, s, t)$ *is a* chronological successor *of* $np' = \delta'(e', s', t')$*, written as* $np \succ_T np'$*, iff* $t > t'$*.*

Definition 4. *A normative position* $np = \delta(e, s, t)$ *is* more salient *than* $np' = \delta'(e', s', t')$*, written as* $np \succ_P np'$*, iff* $s > s'$*.*

In our example, $prh(bid_{ag_1}, 2, 1)$ is more salient and a chronological successor of $per(bid_{ag_1}, 0, 0)$.

3 REGULATED COMPOUND ACTIVITIES

In this paper we have in mind that a performance of a set of actions —subject to some regulation—constitute an activity and that an activity may be composed of several sub-activities which in turn may also be decomposed into other sub-activities. Take the example of a clearinghouse involving different activities outlined in Figure 1. The main activity, trading, involves subactivities —like auctioning, one-to-many negotiation or direct purchasing— that serve the purpose of fixing the conditions for purchasing goods and other activities required for payment and delivery of goods.

Several models and methodologies for MAS (*e.g.*, [1,2,3]) have looked into the notion of compound activity using different names such as performative structure, missions or simply interaction. For our purposes we only need to look into those aspects that relate to the evolution of normative positions within an activity and how these are propagated in the hierarchy.

Each activity behaves like a transition system: it has a state, represented by a set of grounded terms, that changes with the performance of the actions of the agents in the activity. This transition function is partial since not all the actions may occur in all the states of an activity. Norms establish what actions are permitted, forbidden or obligatory (and their effects) in a given state of the activity, defining the transition function of the activity. Normative positions are part of the state of an activity and they also change by the performance of actions and the application of the transition function. Note also that since different activities may be connected —in the sense that what happens in one has effect on the other— when those normative positions in the first change, the normative positions in the other may also change. We will say that the scope of a normative position is the activity where it becomes enabled and all the sub-activities associated with that activity. Finally, recall that conflicts among normative positions could be avoided through the sequential use of criteria (chronological, salience, speciality) that order normative positions. Once these conflicts are resolved, we obtain the set of normative positions that will be applied and propagated to the sub-activities.

All these elements are part of the following definition:

Definition 5. *An* activity state *is a tuple* $q = \langle Ag, Ac, I, N, N^{in}, N^{out}, \tau, O, \Omega \rangle$ *where*

- Ag *is a finite, non-empty set of* agent identifiers*;*
- Ac *is a finite, non-empty set of* action labels*;*
- I *is a finite subset of* grounded terms *that describe the current value of the parameters involved in the activity;*
- N *is the set of normative positions of an activity state* q*;*
- N^{in} *is the set of normative positions propagated from a state of a super-activity to activity state* q*;*
- N^{out} *is the set of applicable normative positions propagated by activity state* q *to the sub-activities;*
- $\tau : Q \times 2^{Ac} \to Q$ *is a partial* transition function *from the set of activity states and sets of actions to the set of activity states* Q*, which defines the state* $\tau(q, ac)$ *that would result by the performance of actions* $ac = \{ac_1, \ldots, ac_n\}$ *from state* q *– note that, as this function is partial, not all the set of actions are possible in all the states;*
- O *is a finite, non-empty set of partial order relations* $\succeq_i$ *in the set of normative positions; and*
- Ω *is a total order relation over* O*.*

Henceforth, we will respectively denote with a q subscript the components of activity state q.

N^{in}_q is the set of normative positions propagated to a activity state q. Algorithm 1, presented in section 4, calculates N^{in} of the super-activities of q prior to use Algorithm 2 to calculate N^{out}_q. Intuitively, the set of inherited normative positions is the union of applicable normative positions of the super-states.

N^{out}_q is the set of applicable normative positions in an activity state q that will be propagated to the sub-activities. It is obtained by removing conflicting normative positions from the union of those that are inherited from super-activities and those that arise from the transition that produces state q. For the removal of conflicting normative positions we use the ordering criteria in O in the sequence established by Ω. To calculate N^{out}_q, we use Algorithm 2 presented in section 4 that applies the meta-ordering Ω of activity state q in $N_q \cup N^{in}_q$ in order to remove the less priority, conflicting normative positions.

Figure 2 shows an example of state of an auctioning activity with 3 agents: an auctioneer, and two buyer agents. Auctioneer made an offer and the buyer can bid for that offer. Buyers have a credit that is decreased when they win an auction. If an agent performs an unsupported bid a sanction of 10 is applied. The set of inherited normative positions (N^{in}) is empty since we assume that auctioning is not part of other activity. The set of applicable normative positions is equal to the set of associated normative positions since there is no conflict. Partial function τ is defined using $\cup_C$ and $\setminus_C$ operators that respectively adds and removes formulae from a set C of the tuple defining an activity state. For instance, $q \cup_N \{obl(pay_{ag_1}, 0, t)\}$ intuitively means that the obligation is added to the set of normative positions N of activity state q. Notice that τ checks the set

$q = \langle Ag, Ac, I, N, N^{in}, N^{out}, \tau, O, \Omega \rangle$ where

- $Ag = \{auct, ag_1, ag_2\}$;
- $Ac = \{offer_{auct}, bid_{ag_1}, bid_{ag_2}, pay_{ag_1}, pay_{ag_2}\}$;
- $I = \left\{ \begin{array}{c} credit(ag_1, 100), credit(ag_2, 50), item(it1), price(it1, 100), \\ decrement(it1, 10), reserveprice(it1, 30) \end{array} \right\}$;
- $N = \{per(bid_{ag_1}, 0, 0), per(bid_{ag_2}, 0, 0)\}$;
- $N^{in} = \emptyset$;
- $N^{out} = \{per(bid_{ag_1}, 0, 0), per(bid_{ag_2}, 0, 0)\}$;
- $$\tau(q, ac) \begin{cases} q \cup_N \{obl(pay_{ag_1}, 0, t)\} & \begin{array}{l} if\ ac = \{bid_{ag_1}\} \text{ and} \\ \quad price(i, p), credit(ag_1, x) \in I_q \text{ and} \\ \quad x \geq p \text{ and } time(t) \end{array} \\ \vdots & \\ \begin{array}{c} (q \setminus_I \{credit(ag_1, x)\}) \cup_I \\ \{credit(ag_1, x - 10)\} \end{array} & \begin{array}{l} if\ ac = \{bid_{ag_1}\} \text{ and} \\ \quad price(i, p), credit(ag_1, x) \in I_q \text{ and} \\ \quad x < p \text{ and } time(t) \end{array} \\ \vdots & \\ \begin{array}{c} ((q \setminus_I \{credit(ag_1, x)\}) \setminus_N \\ \{obl(pay_{ag_1}, s, t)\}) \cup_I \\ \{credit(ag_1, x - p)\} \end{array} & \begin{array}{l} if\ ac = \{pay_{ag_1}\} \text{ and} \\ \quad price(i, p), credit(ag_1, x) \in I_q \text{ and} \\ \quad x \geq p \text{ and } obl(ac, s, t) \in N_q^{out} \end{array} \\ \vdots & \\ q \cup_I \{collision\} & if\ ac = \{bid_{ag_1}, bid_{ag_2}\} \\ \vdots & \end{cases}$$
- $O = \{\succ_T, \succ_P\}$; and
- $\Omega = \{\langle \succ_P, \succ_T \rangle\}$.

Fig. 2. Example of activity state.

of applicable normative positions, since N_q^{out} is supposed to be conflict-free, but changes the set of existing normative positions N_q. Furthermore, there are the two order relations introduced in section 2, as mentioned above salience criterion is preferred over the chronological one.

The scope of norms is established by the scope of the normative positions that the norms enable (or disable). Their scope is the activity (and sub-activities) where their normative positions are associated. When a compound activity has a normative position associated, it is propagated to its sub-activities.

Definition 6. *If an activity A is a* sub-activity *of activity B, denoted $A \ll B$, then there exists at least one state q' of activity B such that its applicable normative positions are a subset of the normative positions propagated to each state q of A. Formally, $A \ll B \implies \forall q \in A, \exists q' \in B : N_{q'}^{out} \subseteq NP_q^{in}$.*

An activity structure defines which sub-activities compose it and which sub-activities compose the former sub-activities by relating the states of activities.

Definition 7. *An* activity structure *is an acyclic directed graph* $H = \langle Q, E \rangle$ *where* Q *is the finite, non-empty set of activity states; and* E *is a finite, non-empty set of edges* $\langle q, q' \rangle$*. If* $\langle q, q' \rangle \in E$ *we say that* q *is a* sub-state *of* q' *and* q' *is a* super-state *of* q*.*

Henceforth, we will denote with a subscript H the components of activity structure H.

Definition 8. *Given two activities* A *and* B*,* A *is a sub-activity of* B *iff all the states of* A *are a sub-state of, at least, one state of* B*. Formally,* $A \ll B \iff \forall q \in A, \exists q' \in B : \langle q, q' \rangle \in E$*.*

3.1 DEONTIC CONFLICTS IN COMPOUND ACTIVITIES

Deontic conflicts can appear either between the set N of normative positions generated in an activity state by the transition function or the set N^{in} of normative positions inherited from other activity states. As expressed in definition 2, two normative positions of the same activity state are in conflict if one is a permission or obligation and the other is a prohibition over the same action than the former. Two normative positions from different activity states are in conflict if one is a permission or obligation and the other is a prohibition over the same action than the former and one of them is associated to a sub-state of the other.

Definition 9. *Given two normative positions* np *and* np'*, respectively pertaining to activity states* q *and* q'*, such that* $np = \delta(a, s, t)$*,* $np' = \delta'(a', s', t')$ *and* $q, q' \in Q_H$ *;* np, np' *are in conflict in an activity structure* H*, denoted* $np \overset{H}{\gg\!\!\!\!\ll} np'$*, iff* $np \gg\!\!\!\!\ll np'$ *and* $q = q'$*; or* $np \gg\!\!\!\!\ll np'$ *and exists a path between* q *and* q' *in* E*.*

By relating activity states in an activity structure, we can adapt the speciality ordering criterion introduced in section 2 to our definition of activity state:

Definition 10. *A normative position* np *is* more specific *than* np' *in activity state* q*, written as* $np \succ_S np'$*, if* $np \in N_q$ *and* $np' \in N_q^{in}$*.*

Given the example of a trading activity composed of two auctioning sub-activities and the activity structure introduced above, we have:

- $N_{trading}^{in} = \emptyset$ since *trading* has no super-state.
- $N_{trading} = \{prh(bid_{ag_1}, 1, 1)\}$ if agent ag_1 made an unsupported bid.
- $N_{trading}^{out} = \{prh(bid_{ag_1}, 1, 1)\}$ since there is no conflict in $N_{trading} \cup N_{trading}^{in}$.
- $N_{auction1}^{in} = \{prh(bid_{ag_1}, 1, 1)\}$ since *auction*1 activity state has only *trading* as super-state and $N_{trading}^{out} = \{prh(bid_{ag_1}, 1, 1)\}$.
- $N_{auction1} = \{per(bid_{ag_1}, 0, 0)\}$, for example, as agents are permitted to bid in auction houses.

Thus, there is a conflict between $N_{auction1}$ and $N_{auction1}^{in}$. Normative position $per(bid_{ag_1}, 0, 0)$ is more specific than $prh(bid_{ag_1}, 1, 1)$ because the former belongs to $N_{auction1}$ and the latter to $N_{auction1}^{in}$. In order to calculate the set of normative positions without conflict, we may use the speciality criterion to remove less priority, conflicting normative positions resulting $N^{out}_{auction1} = \{per(bid_a, 0, 0)\}$.

When the normative positions in an activity state are not conflicting, we will say that this activity state is conflict-free:

Definition 11. *An activity state q is* conflict-free *if there is no conflict among its normative positions (from $N^{in}_q \cup N_q$).*

In the example above, the *trading* activity state is conflict-free since $N^{in}_{trading} \cup N_{trading}$ has no deontic conflict.

When the applicable normative positions in an activity state q (N^{out}_q) are not conflicting, we will say that activity state q is Ω-conflict-free. Ω is the meta-ordering of activity state q used to resolve any deontic conflict.

Definition 12. *An activity state q is Ω-*conflict-free *if the set of applicable normative positions in the activity state (N^{out}_q) is the largest set of normative positions $N^{out}_q \subseteq N^{in}_q \cup N_q$ that is conflict-free after applying meta-ordering Ω of activity state q to resolve deontic conflicts.*

In the example above, *auction*1 activity state is Ω-conflict-free.

On the one hand, there are activity structures without conflicts before applying any method of conflict resolution, we call them *strongly conflict-free.* An activity structure H is strongly conflict-free if there is no conflict in the normative positions of any activity state.

Definition 13. *An activity structure H is* strongly conflict-free *if $\forall q \in Q_H$, $\forall np, np' \in N_q \cup N^{in}_q : \neg(np \overset{H}{\times\!\!\times} np')$.*

On the other hand, there are activity structures without conflicts after applying a method of conflict resolution, we call them *weakly conflict-free.* This property holds when the method of conflict resolution is effective. An activity structure is weakly conflict-free if there is no conflict in the set of applicable normative positions (N^{out}) of any activity state.

Definition 14. *An activity structure H is* weakly conflict-free *if $\forall q \in Q_H$, $\forall np, np' \in N^{out}_q : \neg(np \overset{H}{\times\!\!\times} np')$.*

In the example above, the activity structure that constitutes the trading activity composed of two auctioning activities is not strongly conflict-free because there is a conflict between two normative positions in *auction*1 activity state (in $N_{auction1} \cup N^{in}_{auction1}$). However, it is weakly conflict-free because after applying the Ω meta-ordering, the conflict is resolved and $N^{out}_{auction1}$ includes no conflicting normative positions.

4 MAINTENANCE OF Ω-CONFLICT-FREEDOM

After the norms are applied in each activity state using τ, the set of normative positions associated in each activity state (N) changes. Since the scope of normative positions also include the sub-states of the activity state where the normative position is associated, the set of applicable normative positions should be calculated and propagated to the sub-states.

For that purpose, we introduce in this section PROPAGATE-NPS algorithm that takes as input a list of the leaves of the trees defined by the activity structure and ensures that each activity state in the activity structure is Ω-conflict-free.

Algorithm 1 PROPAGATE-NPS($leaves, h$)

```
Require: $leaves$ is a list of the leaves of each tree of the activity structure $h = \langle Q, E \rangle$
Ensure: Every $N^{out}$ is conflict-free
 1: for all $leaf \in leaves$ do
 2:   if $N^{out}_{leaf} = null$ then {First execution for activity state $leaf$}
 3:     $parents \leftarrow \{q' \mid \langle leaf, q' \rangle \in E\}$
 4:     $l \leftarrow \emptyset$
 5:     if $parents \neq \emptyset$ then
 6:       PROPAGATE-NPS($parents, h$)
 7:       for all $parent \in parents$ do {Append $N^{out}$ of parents}
 8:         $l \leftarrow l \cup N^{out}_{parent}$
 9:       end for
10:     end if
11:     $N^{in}_{leaf} \leftarrow l$
12:     $N^{out}_{leaf} \leftarrow$ GET-CONFLICTFREENPS($leaf$)
13:   end if
14: end for
15: return $leaves$
```

In algorithm 1, for each leaf of the activity structure, if N^{out} has not been calculated yet (line 2), we apply recursively PROPAGATE-NPS to the parents (if they exist) (line 6) and we gather the list of all the normative positions inherited by the current activity state and update N^{in} (line 12). We set to N^{out}, the result of the algorithm GET-CONFLICTFREENPS applied to the updated activity state (line 13).

When normative positions are propagated, the set of applicable normative positions of an activity state should be calculated. For that purpose, Algorithm 2 returns this set ensuring that it is conflict-free. In algorithm 2, for each normative position np, we gather in s, by calling GET-SETINCONFLICT, a list of normative positions including np and the ones in conflict with np (line 4). If there is at least one normative position in conflict, then we call GET-PRIORITYNP to resolve the conflict and get the highest priority normative position (line 6). This normative position is added to the result list anp (line 8). Otherwise, np is added to the result list in anp (line 11).

Algorithm 2 Get-ConflictFreeNPs(q)

```
Require: $q = \langle Ag_q, Ac_q, I_q, N_q, N_q^{in}, N_q^{out}, \tau_q, O_q, \Omega_q \rangle$
Ensure: $anp$ is the list of applicable normative positions
$anp \leftarrow \emptyset$
$nps \leftarrow N_q \cup N_q^{in}$
for all $np \in nps$ do
    $s \leftarrow$ Get-SetInConflict($np, nps$)
    if Length($s$) $> 1$ then
        $pnp \leftarrow$ Get-PriorityNP($\Omega_q, s$)
        if $pnp \notin anp$ then
            $anp \leftarrow anp \cup pnp$
        end if
    else if Length($s$) $= 1$ then
        $anp \leftarrow anp \cup np$
    end if
end for
return anp
```

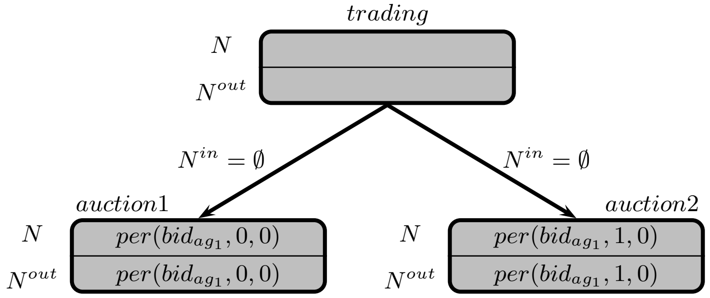
Although different activity states can share super-states, Algorithm 1 calculates N^{out} of each activity state only once. Addition and removal of normative positions from a activity state require that N^{out} for that activity state and its sub-states are recursively set to null using Algorithm 3.

Algorithm 3 Clear-ANP(q, h)

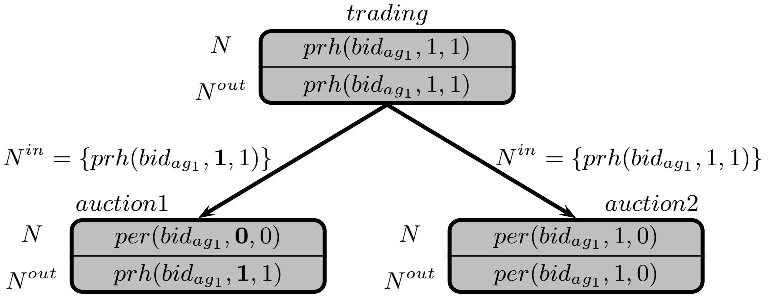
```
Require: $q = \langle Ag_q, Ac_q, I_q, N_q, N_q^{in}, N_q^{out}, \tau_q, O_q, \Omega_q \rangle$ and $h = \langle Q, E \rangle$
Ensure: For $q$ and its sub-states, $N^{out} = null$
$N_q^{out} \leftarrow null$
$children \leftarrow \{q' \mid \langle q', q \rangle \in E\}$
for all $child \in children$ do
    Clear-ANP($child$)
end for
```

5 EXAMPLE

In this section, we introduce an example of a regulated compound activity. Picture a set of auctioning activities, regulated by their own norms, that constitute a trading activity. At the trading activity level there is a norm stating that a buyer that makes an unsupported bid in a auctioning activity will be banned to bid in any of the auctioning activities except in the auctioning activity called *auction*2. The meta-ordering to be applied will be $(\succ_P) \prec (\succ_S) \prec (\succ_T)$: the salience criterion prevails over the speciality and the chronological ones, while the speciality criterion prevails over the chronological one.



(a) activity states before an unsupported bid



(b) activity states after an unsupported bid

Fig. 3. Example of the normative state of the activity structure

Figure 3 shows an example of execution for a trading activity compound of two auctioning activities and a buyer agent. Figure 3(a) illustrates the state of the activities prior to any unsupported bid. Activity state *trading* has no normative position associated. Thus, its set of applicable normative positions is empty. Activity state *auction*1 only has associated a permission with normal priority (salience 0). Since N^{out} of the super-state is empty, its N^{out} only contains the associated permission. Activity state *auction*2 only has associated a permission with higher priority (salience 1). Since the N^{out} of the super-state is empty, its N^{out} only contains the associated permission.

Figure 3(b) shows the state of the activities after an unsupported bid performed by agent ag_1. Activity state *trading* has associated the prohibition (with salience 1) for ag_1 to bid. Since the *trading* state has no super-states, its N^{out} is equal to its associated normative positions, i.e., the prohibition. Recall that activity state *auction*1 only has associated a permission with normal priority (salience 0). Since the prohibition in N^{in} (inherited from *trading* activity state) has higher salience, it will belong to the N^{out} of *auction*1 state. Recall that *auction*2 state only has associated a permission with salience 1. Since the prohibition in N^{in} (inherited from *trading* state) has the same salience, the speciality criterion is applied. The permission will belong to the N^{out} of *auction*2 state because it belongs to N, i.e., it is associated to the sub-state.

6 RELATED WORK

There are many works in deontic conflicts (*e.g.*, [8,9,10,11,12]) from a logical point of view but there are few works that implement computational strategies to solve deontic conflicts in the area of multi-agent systems. Two developments are related quite directly with the problems we face in our proposal [13,14].

In [13], the authors present an agent architecture where obligations, permissions and prohibitions can be added to the agents' plans. Contrary to our work, in which deontic conflicts appear when defining sub-activities, deontic conflicts appear in this work when the agent has to adopt non-hierarchical norms. They (explicitly or implicitly) associate norms to Instantiation Graphs which represent an action or state declaration as a hierarchy of all its possible forms of partial instantiation of variables. Thus, norms are also ordered: explicit norms override implicit norms; and new norms override old norms. Although they consider deontic conflicts, they only adopt an agent-centred stance.

In [14], the author proposes the use of $RuleML$ for representing business contracts. The underpinning of the proposal is the use of Defeasible Logic (DL) as the inferential mechanism for $RuleML$. The primary use of DL in that work is the resolution of conflicts that might arise from clauses of a contract. DL analyses the conditions laid down by each rule in the contract, identifies the possible conflicts that may be triggered and uses the priorities defined over the rules to eventually resolve a conflict. By using DL, a normative position receive different priority depending on the antecedents of the rule and not on the normative position by itself. In contrast, our priorities are defined at the normative position level, i.e., we assign a priority to each normative position.

7 FINAL REMARKS

We took an unconventional approach to a complex problem and in this paper we made many simplifying assumptions that we intend to relax in future work. For the moment we wanted to keep our framework simple so that we could explore the main components of the problem. We also wanted to keep it concrete enough so that it could be applied to real organisations and because of that aim we wanted to profit from implemented systems that are already available to deal with regulated simple activities.

In spite of the austerity of this proposal, it is evident that most intuitions we have explored here are prone to a serious logical treatment. Conflict resolution in this paper has been limited to a total ordering of normative positions. It is true that this type of resolution may be adequate in some conflicts and some activities. However, we realise that this issue may be treated in other interesting ways accommodating other culturally accepted conflict resolution mechanisms like negotiation, arbitration or argumentation. Likewise, the current conflict resolution components of the framework could be revisited to incorporate other pertinent normative aspects like peer to peer conflict settlements, contract breaches and blame assignment, enforcement policies, etc.

To define normative positions, in this paper we use an action identifier a that hides too much information. Here we only wanted to be able to decide whether two actions are the same or not. However, that makes the crude handling of normative positions to be simplistic stand-in for commitments. In a system that does serious commitment management, action identifiers will very likely be terms that adequately reflect the state the activity and the organisation at the time of establishing the commitment: in particular, who are the agents involved in the commitment and what are the specific current values of the parameters involved in the definition of the commitment and its fulfilment or up-keeping.

In a similar vein, we have assumed a somewhat obscure notion of activity state because we have hidden many significant elements inside the "transition function". One option we have at hand to make our notion of transition functions clear is to use the notion of performative structures [1]. In that light, the transition function correspond to their speech-acts labelled finite-states machines of a scene together with the scene transitions. Another option is to think of activities as logical theories with a deduction mechanism. In that case, transitions occur when a new action (or possibly concurrent actions) is added to the theory and its consequences deducted. In both cases, performative structures and logical theories, for each transition we still need to keep track of the starting and terminating states of the activity. For that purpose, we may take from [1] the notion of scene state and institutional state and extend them by including the state of the normative positions in the activity and in the compound activities. As part of the state of an activity we need to take into account those commitments that are active, but also their relevance to the actions that take place so that their propagation is correct (sound and complete).

The propagation of normative positions that we discuss in this paper is hand-wired and fix on top of the activities hierarchy. The paths of commitment propagation should be an inherent outcome of the way activities are regulated and combined to constitute the regulated environment. Furthermore, although it may be rather natural to assume nested hierarchies of activities, the express assumption of acyclicity is questionable from an applications point of view, no matter how convenient it may be for formal and algorithmic reasons. In this respect, it looks attractive to distinguish between propagation links among activities that are established through the flow transitions designed into the compound activities on one hand and propagation between states whenever an action takes place anywhere in the organisation and for that purpose a notion of pertinence would be welcome.

We have taken care to make our proposal compatible with the model from [1] and as such it constitutes an extension of that model. In that language we can say that our activities correspond to scenes, performative structures, and nested normative structures. Because of this last possibility, we need an extended notion of transition to handle connections between all of these that produces non-nested compound activities. In addition, this richer activity composition leads us to consider an enriched notion of transition that deals in a smooth way with commitment propagation. We are also interested in profiting from expressive extensions to that model so that conventions may be stated not only as transition graphs but as normative expressions of good expressive power endowed also with a deduction mechanism. Our proposal, as it stands, should work properly with our recent production rules extensions [15].

In this prospective paper, we have followed an unconventional approach to the problem of managing dynamic commitment-making in regulated agent systems. We think that the problem is a fundamental one for regulated environments where the autonomy of participants is an essential ingredient. Although we are aware that what we propose here is far from being a solution to that problem, we acknowledge that the approach has brought to light many challenging questions that we believe deserve further analysis.

Acknowledgements

This work was partially funded by the Spanish Science and Technology Ministry as part of the Web-i-2 project (TIC-2003-08763-C02-00). García-Camino enjoys an I3P grant from the Spanish Scientific Research Council (CSIC).

References

1. Arcos, J.L., Esteva, M., Noriega, P., Rodríguez-Aguilar, J.A., Sierra, C.: Engineering open environments with electronic institutions. Engineering Applications of Artificial Intelligence **18**(2) (2005) 191–204
2. Hannoun, M., Boissier, O., Sichman, J.S., Sayettat, C.: MOISE: An organizational model for multi-agent systems. In Monard, M.C., Sichman, J.S., eds.: Advances in Artificial Intelligence. Volume 1952 of LNAI. Springer-Verlag (2000)
3. Dignum, V.: A model for organizational interaction: based on agents, founded in logic. PhD thesis, Utrecht University (2003)
4. Yolum, P., Singh, M.: Flexible protocol specification and execution: Applying event calculus planning using commitments. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02). (2002)
5. Sergot, M.: A Computational Theory of Normative Positions. ACM Trans. Comput. Logic **2**(4) (2001) 581–622
6. Pacheco, O., Carmo, J.: A role based model for the normative specification of organized collective agency and agents interaction. Autonomus Agents and Multi-Agent Systems **6** (2003) 145–184
7. García-Camino, A., Rodríguez-Aguilar, J.A., Sierra, C., Vasconcelos, W.: Norm-Oriented Programming of Electronic Institutions: A Rule-based Approach. In: Coordination, Organization, Institutions and Norms in agent systems (COIN'06) in Fifth International Joint Conference on Autonomous Agents and Multiagent Systems. (AAMAS'06). (2006)

8. Sartor, G.: Normative conflicts in legal reasoning. Artificial Intelligence and Law **1**(2-3) (1992) 209–235
9. : International Workshop in Deontic Logic in Computer Science. DEON. (1991-2006)
10. Cholvy, L., Cuppens, F.: Solving normative conflicts by merging roles. In: Fifth International Conference on Artificial Intelligence and Law, Washington, USA (1995)
11. Cholvy, L., Cuppens, F.: Reasoning about norms provided by conflicting regulations. In McNamara, P., Prakken, H., eds.: Norms, Logics and Information Systems. New Studies in Deontic Logic and Computer Science. Volume 49 of Frontiers in Artificial Intelligence and Applications. IOS Press, Washington, USA (1999) 247–262
12. Boella, G., van der Torre, L.: Permission and Obligations in Hierarchical Normative Systems. In: Procs. 8th Int'l Conf. in AI & Law (ICAIL'03), Edinburgh, ACM (2003)
13. Kollingbaum, M., Norman, T.: Strategies for resolving norm conflict in practical reasoning. In: ECAI Workshop Coordination in Emergent Agent Societies 2004. (2004)
14. Governatori, G.: Representing business contracts in $RuleML$. International Journal of Cooperative Information Systems **14**(2-3) (2005) 181–216
15. García-Camino, A., Noriega, P., Rodríguez-Aguilar, J.A.: Implementing Norms in Electronic Institutions. In: Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'05), Utrecht, The Nederlands (2005) 667–673