Normative Conflict Management with Default Logic

Georgios K. Giannikis and Aspassia Daskalopulu

Department of Computer and Communications Engineering.
University of Thessaly,
38221 Volos, Greece
{ggiannik,aspassia}@inf.uth.gr

Abstract. The analysis, representation and management of normative conflicts has been the focus of much research in recent years, from a variety of perspectives, such as distributed systems management, legal reasoning and argumentation, and multi-agent interaction. This report is concerned with normative conflicts that arise for agents engaging in electronic contracting, and presents a set of primitive conflict patterns. The report examines other analyses of normative conflicts and shows how these may be seen as instances of these primitives. It proposes a representation of e-contracts in Default Logic, which facilitates the detection, prediction and resolution of such normative conflicts.

Keywords. E-contracts, Conflict Management, Default Logic

February 2007

Table of Contents

1 Introduction

The analysis, representation and management of normative conflicts has been the focus of much research in recent years, from a variety of perspectives, such as distributed systems management, legal reasoning and argumentation, and multi-agent interaction.

In the early nineties, Sartor's [29] and Horty's [15] work on normative conflicts set the theoretical basis for conflict management. According to Sartor [29] a conflict arises when "(possibly) valid norms establish incompatible qualifications for the same concrete state". The corner stone in this approach is a *norm* set. This may be either *inconsistent*, if a contradiction is logically derivable from it, or *potentially inconsistent*, if it may lead to contradiction in an upcoming state. In similar spirit Horty in [15, 16] addresses moral conflicts: an agent is in moral conflict if it ought to do an action A and, at the same time, it ought to do another action B, but it is impossible to do both.

This report is concerned with normative conflicts that arise for agents engaging in electronic contracting, within an electronic marketplace. We seek appropriate characterizations for normative conflicts that would enable the detection (and resolution) of actual inconsistencies, as well as the prediction (and avoidance) of possible inconsistencies.

Many approaches have emerged during the past decade for the analysis and management of normative conflicts. Most of them use the deontic notions of obligation, permission and prohibition in their analyses and propose that conflict resolution may be achieved through the ascription of priorities to norms, based on some criterion (e.g. norms may be hierarchically organized through some legal rule, or through some domain-specific policy, or through some hierarchy of roles). Moffett *et al.* [24], Lupu *et al.* [21] and Dunlop *et al.* [8, 9] address conflicts from the Distributed Systems Management viewpoint and view policies as a way to determine and influence management behaviour. Cholvy *et al.* [5, 6] view normative conflicts as the result of role conflict and propose a solution based on hierarchies of roles. Broersen *et al.* [4] deal with conflicts that arise between an agent's mental attributes such as beliefs, obligations, intensions and desires. Kowalski [19] considers normative conflicts that arise as a result of conflicting goals and presents an approach that unifies logic and decision theory. Finally, Kollingbaum *et al.* [18] focus on practical reasoning agents and use instantiation graphs to detect conflicts.

In this work we investigate an alternative approach, in which we use Reiter's Default Logic (DfL) [27]. We propose the representation of contractual norms as default rules, so that we may reason non-monotonically with them. We identify a set of primitive patterns for normative conflicts and show how the conflicts identified by other researchers may be seen as instances of these primitives. We also identify some patterns of normative conflict that have not been identified in other proposals. Finally, we argue that the representation of contractual norms as default rules facilitates both conflict detection (and prediction) and resolution (and avoidance).

2 Preliminaries

For the purposes of illustration consider an electronic marketplace, populated by software agents that establish and perform e-contracts on behalf of some real world parties. Let the set *Agents*={*Agent*1, *Agent*2, *Agent*3,.....} denote distinct identifiers for the various agents, and the set *Roles*={*RA*, *WA*, *MA*, *CA*, ... } denote distinct roles that agents may assume in the e-market (where *RA*, *WA*, *MA*, *CA* denote ***retailer***, ***wholesaler***, ***mediator*** and ***carrier*** respectively).

Consider a two-party business transaction. *Agent*1 that acts as a retailer orders some goods from the wholesaler *Agent*3. The terms of the agreement between these two agents are: *Agent*3 should see to it that the goods be delivered to *Agent*1 within 10 days from commencement (e.g., the date that the order takes place). *Agent*1, in turn, should see to it that payment be made within 21 days from the date it receives the goods. If *Agent*3 does not deliver on time, then a fixed amount is to be deducted from the original price of the goods for each day of delay and it should see to it that delivery be made by a new deadline. If *Agent*1 does not perform payment on time, then a fixed amount is to be added to the original price of the goods for each day of delay and it should see to it that payment be made by a new deadline.

Following [7], we may take an informal, process view of the business transaction that is regulated by the agreement. Each state offers a (possibly partial) description of the factual and normative propositions that hold true. A transition between states corresponds to an event that takes place, i.e. an action that one of the parties performs or omits to perform. Normative propositions of the form:

NN(*agent*1, *role*1, *action*, *agent*2, *role*2)

express that *agent*1 that acts as *role*1 is in legal relation *NN* towards *agent*2 that acts as *role*2 to perform *action*, where *NN* may be *Obligation*, *Prohibition*, *Permission* and legal *Power*.

We use Reiter's Default Logic [27] to represent the norms of an agreement as default rules. A default rule has the form $P{:}J_1,J_2,...J_n/C$, where *P* is the ***prerequisite***, $J=\{J_1,J_2,...\ J_n\}$ is a set of ***justifications*** and *C* is the derived ***consequent***. If *J* coincides with *C*, the default rule is called *normal*. The semantics of a default rule is: If *P* holds and the assumption *J* is consistent with our current knowledge, then *C* may be inferred.

For instance, the following default rule expresses that if an order from *Agent*1 (acting as a retailer) towards *Agent*3 (acting as a wholesaler) holds, and it is consistent to assume that *Agent*1 will become a regular client, then we may infer that *Agent*3 is legally obliged towards *Agent*1 to perform delivery:

$$\frac{Order(Agent1, RA, Agent3, WA)\ :\ BecomeRegularClient(Agent1)}{Obligation(Agent3, WA, Delivery, Agent1, RA)}$$

A Default Theory (DfT) is a pair of the form (W, D), where W is a set of closed formulae that hold, and D is a set of defaults. Rules may be used to compute *extensions* E of the default theory. A rule is applicable to a set of formulae $W \subseteq E$ if and only if $P \in E$ and $\neg J_1, \ldots, \neg J_n \notin E$. We consider grounded DfTs and we derive extensions in the manner presented in [2], i.e. by maintaining consistent sets of formulae. This derivation may be conducted in stepwise manner.

An agent that engages in some agreement-governed transaction essentially reasons with a default theory. At each time point during the business transaction the agent attempts to compute the extensions of its current DfT. A normative conflict may be detected either between multiple extensions or between some extension and the current knowledge (W) of the agent. Where a conflict is detected between multiple extensions, the latter represent alternative futures for the agent. Where a conflict is detected between an extension and the current knowledge of the agent, it represents a state in which some normative violation occurs. The role of conflict detection is, thus, to assist an agent to choose a course of action so that violations may be predicted.

3 Conflict Detection

The first step of conflict management involves the detection of conflicts. To this end, in section 3.1, we identify primitive patterns of normative conflict that may be spotted during the derivation of extensions of the default theory representation of an agreement. In section 3.2 we discuss other analyses of normative conflicts and show how these may be seen as instances of the primitive patterns.

3.1 Primitive Patterns of Normative Conflicts

In what follows we use *Obligation*, *Permission*, *Prohibition* and *Power* as predicates that express normative relations between agents. We do not employ the axiomatization of any particular system of Deontic logic; specifically, we do not employ the axiomatization of Standard Deontic Logic, in which these notions are modeled as operators that are inter-defined.

A. Conflict between a normative notion (NN) and its negation. The general pattern is:

$$NN(agent1, role1, action, agent2, role2)$$

$$\neg NN(agent1, role1, action, agent2, role2)$$

This is the common syntactical conflict that arises when an agent has contradictory knowledge. All other approaches, without any exception, refer to this type of conflict. In policy-based approaches, when the normative notion is *Obligation* it is called *positive-negative conflict of modalities* [24]. This type of conflict never actually arises in our representation, where norms are represented as defaults, because the derivation of extensions preserves consistency. It may, however, be identified as a potential conflict, when multiple extensions are computed.

B. Conflict between the prohibition to perform an action and the simultaneous permission or obligation to perform the same action. Once again, all previous research approaches refer to this type of conflict. In [24] and [21] these conflicts are called *conflicts between authority policies* (sub-pattern B1: Prohibition vs Permission) and *conflict between authority and imperatival policies* (sub-pattern B2: Prohibition vs Obligation) respectively.

Consider, for instance, the following default theory (W, D) where:

$W=\{Order(Agent1, RA, Agent3, WA)\}$

and $D=\{$

$$\frac{Order(Agent1, RA, Agent3, WA) \; : \; WellKnownDebtor(Agent1)}{Prohibition(Agent3, WA, Delivery, Agent1, RA)}$$

$$\frac{Order(Agent1, RA, Agent3, WA) \; : \; Permission(Agent3, WA, Delivery, Agent1, RA)}{Permission(Agent3, WA, Delivery, Agent1, RA)} \quad \}$$

The first default denotes that if an order from *Agent*1 (acting as retailer) towards *Agent*3 (acting as wholesaler) holds, and it is consistent to assume that *Agent*1 is related to a well known debtor then we may infer that *Agent*3 is prohibited to perform delivery. Similarly, the second default expresses that if an order from *Agent*1 towards *Agent*3 holds, and it is consistent to assume that *Agent*3 is permitted to perform delivery, then we may infer that *Agent*3 is permitted to perform delivery towards *Agent*1. *Agent*3 may find itself in a conflicting state after applying the two defaults sequentially. We denote this type of conflict as B1.

In the same spirit, consider the following DfT:

$W=\{Order(Agent1, RA, Agent3, WA)\}$

and $D=\{$

$$\frac{Order(Agent1, RA, Agent3, WA) \; : \; WellKnownDebtor(Agent1)}{}$$

$$\frac{}{Prohibition(Agent3, WA, Delivery, Agent1, RA)}$$

$$\frac{Order(Agent1, RA, Agent3, WA) : Obligation(Agent3, WA, Delivery, Agent1, RA)}{Obligation(Agent3, WA, Delivery, Agent1, RA)}\}$$

Once again *Agent*3 is in conflict. We denote this conflict between Prohibition and Obligation as B2.

C. Conflict between an obligation to perform *action* and the simultaneous obligation or permission to perform *¬action*. Here *¬action* denotes a negative action, and the issue of representing negative actions has concerned researchers (e.g. [28] regards them as actions that do not lead to the successful fulfillment of a norm). We have not developed special semantics for the representation of negative actions; we merely regard such expressions as denoting either performance of some action other than the negative one, or as idleness (non performance of any action). This case arises, also, in Lee [20] and Abrahams [1] who use the term *Waive*.

For example consider the following DfT where:

$W=\{Order(Agent1, RA, Agent3, WA)\}$

and $D=\{$

$$\frac{Order(Agent1, RA, Agent3, WA) : BecomeRegularClient(Agent1)}{Obligation(Agent3, WA, Delivery, Agent1, RA)}$$

$$\frac{Order(Agent1, RA, Agent3, WA) : WellKnownDebtor(Agent1)}{Obligation(Agent3, WA, \neg Delivery, Agent1, RA)}\}$$

D. Conflict between the power to perform an action and the simultaneous prohibition to perform the same action. This type of conflict is also referred in [22, 17, 1].

For instance consider the following DfT:

$W=\{Order(Agent1, RA, Agent3, WA)\}$

and $D=\{$

$$\frac{Order(Agent1, RA, Agent3, WA) : Power(Agent3, WA, Delivery, Agent1, RA)}{Power(Agent3, WA, Delivery, Agent1, RA)}$$

$$\frac{Order(Agent1, RA, Agent3, WA) : WellKnownDebtor(Agent1)}{Prohibition(Agent3, WA, Delivery, Agent1, RA)} \quad \}$$

E. Conflict between two obligatory distinct actions, where it is impossible to do both at the same time. This corresponds to Horty's moral dilemma [15].

For instance consider the following DfT where:

$W=\{Order(Agent1, RA, Agent3, WA),$

$Order(Agent2, RA, Agent3, WA),$

no simultaneous performance of actions is possible$\}$

and $D=\{$

$$\frac{Order(Agent1, RA, Agent3, WA) : BecomeRegularClient(Agent1)}{Obligation(Agent3, WA, Delivery1, Agent1, RA)}$$

$$\frac{Order(Agent2, RA, Agent3, WA, T1) : IsRegularClient(Agent1)}{Obligation(Agent3, WA, Delivery2, Agent2, RA)} \quad \}$$

*Agent*3 bears two obligations that cannot be simultaneously satisfied.

F. Conflict between an obligation and the negation of the agent's permission or power to perform it. The negation of an agent's permission/power to perform an action may be explicitly derived from the agent's knowledge base (sub-pattern F1) or it may be derived from a possibly incomplete knowledge base, through the absence of explicit information (sub-pattern F2).

For instance consider the following default theory where:

$W=\{Order(Agent1, RA, Agent3, WA)\}$

and $D=\{$

$$\frac{Order(Agent1, RA, Agent3, WA) : BecomeRegularClient(Agent1)}{Obligation(Agent3, WA, Delivery, Agent1, RA)}$$

$$\frac{Order(Agent1, RA, Agent3, WA) : WellKnownDebtor(Agent1)}{\neg Permission(Agent3, WA, Delivery, Agent1, RA)} \quad \}$$

Now consider a DfT that contains the first of the defaults above and in place of the second, the following

$$\frac{Order(Agent1, RA, Agent3, WA) : \neg Permission(Agent3, WA, Delivery, Agent1, RA)}{\neg Permission(Agent3, WA, Delivery, Agent1, RA)}$$

If the agent's knowledge base does not contain an explicit permission, then the justification of this default will be satisfied.

3.2 Other Analyses of Normative Conflicts

In this section we review some of the main ideas that other researchers have proposed in their analyses of normative conflict and discuss how these may be regarded as instantiations of the primitive patterns presented in the previous section. We also present some additional cases of normative conflict that we have identified, which may also be seen as instances of the primitive patterns. Although all the patterns discussed in this section may be regarded as special cases of the primitive patterns we introduced, they merit a separate discussion because they contain additional information that may be useful for efficient conflict resolution. We do not show the entire DfT, where it is clear that the conflicting norms arise as a result of distinct default rules.

3.2.1 Policy-based Conflicts

G. Intra-policy conflicts. Dunlop *et al.* [8] refer to an *internal policy conflict*, when contradictory policies are assigned to a single role. A policy in their approach corresponds to what we call a single norm.

   Consider, for example, the two distinct obligations of *Agent*3 (a wholesaler) to perform delivery towards two distinct retailers (*Agent*1 and *Agent*2).

$$Obligation(Agent3, WA, Delivery, Agent1, RA)$$

$$Obligation(Agent3, WA, Delivery, Agent2, RA)$$

   The conflict arises from the fact that contradictory policies are assigned to *Agent*3 when acting as wholesaler. Apparently, this specific case can be mapped onto pattern E. In the same manner, other examples of this kind may be seen as instances of other primitive patterns.

H. Inter-policy conflicts. Dunlop *et al.* [8] refer to an *external policy conflict*, when an agent simultaneously assumes different roles that contradict "in co-existence".

   Consider, for example, that when *Agent*3 acts as a wholesaler it is obliged to perform delivery towards *Agent*1 while when it acts as a mediator it is prohibited to perform the same action.

$$Obligation(Agent3, WA, Delivery, Agent1, RA)$$

$$Prohibition(Agent3, MA, Delivery, Agent1, RA)$$

   This specific example can be mapped onto pattern B2.

3.2.2 Role-based Conflicts

I. Intra-role conflicts. Cholvy *et al.* [5], consider conflicts only among different roles. In their approach a role is defined through a set of consistent norms. We believe that for a variety of applications it is not realistic to insist on consistent role definitions, and thus we accept intra-role conflicts. Typical examples of this kind of conflict are *authority conflicts* [24] and conflicts that are related with the notion of power.

   Consider the case where *Agent*3 who acts as a wholesaler is both permitted and prohibited to perform delivery towards the retailer *Agent*1. This inconsistency may arise depending on the assumptions that are made, such as the ones presented earlier on the relation of the retailer with a well known debtor.

$$Permission(Agent3, WA, Delivery, Agent1, RA)$$

$$Prohibition(Agent3, WA, Delivery, Agent1, RA)$$

Apparently, this case can be mapped onto pattern B1.

J. Inter-role conflicts. Cholvy *et al.* [5] and Dunlop *et al.* [8] identify an inter-role conflict when contradictory norms arise as a result of multiple roles being assigned to an agent.

For example, when *Agent*3 acts as a carrier it is obligatory to perform delivery. If, at the same time, the same agent assumes the role of wholesaler, then such delivery is not obligatory.

$$Obligation(Agent3, CA, Delivery, Agent1, RA)$$

$$\neg Obligation(Agent3, WA, Delivery, Agent1, RA)$$

This case can be mapped onto pattern A.

Obviously cases G and I, as well as cases H and J are conceptually related. The respective authors use the terms "policy" and "role" differently, and the only reason for discussing them separately is to facilitate comparison.

3.2.3 Conflicts related to Interest/Duty

K. Conflicts of interest. Moffett *et al.* in [24] define *conflicts of interest* as the situation where "the same subject can perform management tasks on two different sets of targets". This type of conflict can be seen as an instance of J (inter-role conflict), or H (inter-policy conflict) or the primitive E (conflict between two obligations).

L. Conflicts of Duty. Moffett *et al.* in [24] and later Lupu *et al.* in [21] define *conflicts of duties* and *application specific conflicts* respectively. They refer to situations where the same agent should not be allowed to perform two distinct actions (e.g. the same agent should not be allowed both to enter a payment and to sign the payment cheque). Such conflicts may be seen as instances of J (inter-role conflict), or H (inter-policy conflict) or E (conflict between two obligations).

3.2.4 Exceptions

M. Exceptions. This type of conflict arises generally in norm-governed systems. As Sartor [29] notes such conflicts emerge when "exceptions to norms state that particular norms, unambiguously identified, do not apply in a given situation".

Consider the following DfT where the retailer *Agent*1 who holds a discount card orders goods from the wholesaler *Agent*3. Based on the first default the retailer gets a 10% discount due to the discount card. On the other hand, based on the second default the retailer should get a 20% discount because it places an order during the sales period.

$W=\{Order(Agent1, RA, Agent3, WA)\}$

and $D=\{$

$$\frac{Order(Agent1, RA, Agent3, WA) \wedge HasDiscountCard(Agent1, RA) : Obligation(Agent3, WA, Discount10\%, Agent1, RA)}{Obligation(Agent3, WA, Discount10\%, Agent1, RA)}$$

$$\frac{Order(Agent1, RA, Agent3, WA) \wedge SalesPeriod : Obligation(Agent3, WA, Discount20\%, Agent1, RA)}{Obligation(Agent3, WA, Discount20\%, Agent1, RA)} \}$$

3.2.5 Temporal Normative Conflicts

Dunlop *et al.* [8] present a temporal logic based approach for the detection of normative conflicts. In this section we present briefly a modification of our representation of agreements in DfL, which takes into account the *external* time of a norm (i.e. the time at which it comes into force) and the *internal* time of a norm (i.e. the time stipulated for its satisfaction, its deadline) (cf. [23]). A formula of the form:

$$NN(agent1, role1, action, time2, agent2, role2, time1)$$

denotes that at time point *time*1 *agent*1 (acting as *role*1) is in legal relation *NN* towards *agent2* (acting as *role*2) to perform *action by time*2.

Now, we may discuss normative conflicts of the types described by the primitive patterns B-F, in a temporal setting. For the purposes of illustration consider the primitive pattern E, in which the following norms are in conflict:

$$Obligation(Agent3, WA, Delivery, IT1, Agent1, RA, ET1)$$

$$Obligation(Agent3, WA, Delivery, IT2, Agent2, RA, ET2)$$

where *IT*1, and *ET*1 are the internal/external time points for the first norm, and *IT*2, *ET*2 are the internal/external time points for the second norm. Temporally well formed norms are those whose internal time is subsequent to their external time, so each normative proposition corresponds to an interval; the intervals for the example we use here are $\Delta T1=[ET1, IT1]$ and $\Delta T2=[ET2, IT2]$.

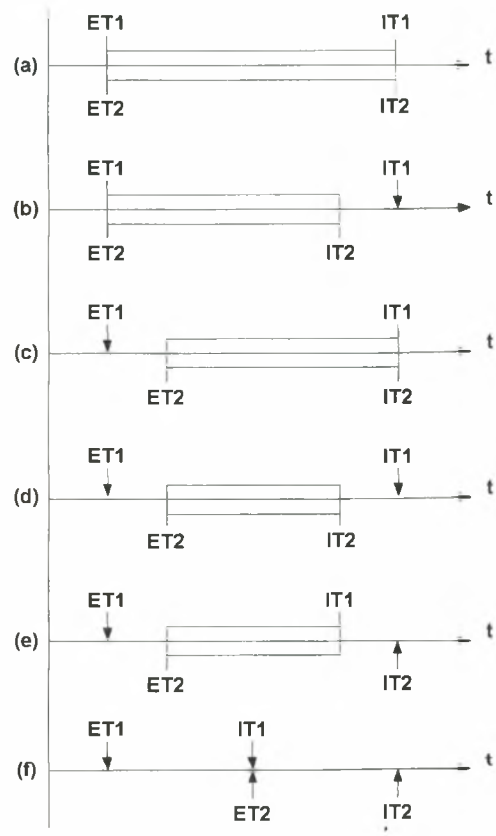

Figure 1 Time interval-based conflicts

N. A conflict arises in the following situations (these are depicted as shadowed in figure 1):

(i) $ET1 = ET2$ *and* $IT1 = IT2$: when $\Delta T1$ coincides with $\Delta T2$ (Figure 1(a)).

(ii) $ET1 \leq ET2 < IT2 \leq IT1$: when $\Delta T1$ fully overlaps $\Delta T2$ (Figure 1(b), 1(c) and 1(d)).

(iii) $ET1 < ET2 < IT1 < IT2$: when $\Delta T1$ partially overlaps $\Delta T2$ (Figure 1(e)).

(iv) $IT1 = ET2$: when $\Delta T1$ meets $\Delta T2$ (Figure 1(f)). This conflict holds only at time point $IT1 = ET2$.

Note that for completeness, one should also consider the symmetrical cases.

3.2.6 Additional Patterns

Here are some additional cases of normative conflict that we have identified, which are not discussed already in the existing literature. We mention them separately because, although they may be reduced to the primitive patterns, there is additional information that may be exploited to facilitate conflict resolution.

O. Type of action-based conflicts. A common feature of e-contracts is the so called Contrary-to-Duty structures [26]. An agent's contractual obligations may be divided in two types. *Prima facie* obligations that concern the performance of actions that are in principle stipulated by the agreement and secondary obligations that concern the performance of reparatory actions; the latter apply only when violations of *prima facie* obligations happen.

   An agent may, thus, bear two distinct obligations (for instance of the kind described by G or E), where one is primary and the other is secondary. This qualification may be helpful in conflict resolution, as will be discussed in section 4.

P. Agreement-based conflicts. An agent may find itself in a conflicting state because it is engaged in multiple contracts. For instance a wholesaler may be obliged to perform two distinct deliveries to two distinct retailers as dictated by two distinct agreements. This situation may be regarded as the generalization of patterns G and E discussed earlier, because in this case the important information is the distinction between the contracts. The additional information that the two norms stem from two agreements, may be exploited for the purposes of conflict resolution. Note that this conflict pattern is different form the one presented in [14]. The key notion here is the different contracts an agent has to comply with. Different contracts may be established towards different agents or even towards the same agent.

Q. Conflicts between assumptions and knowledge. A conflict may arise not only as a result of an agent's explicit knowledge but also between its knowledge and its current assumptions.

   For example, according to the following DfT the prohibition that derives from the second default contradicts not only with obligation that derives from the first default, but also with the assumption of the first default (power).

   $W=\{Order(Agent1, RA, Agent3, WA)\}$

   and $D=\{$

$$\frac{Order(Agent1, RA, Agent3, WA) : Power(Agent3, WA, Delivery, Agent1, RA)}{Obligation(Agent3, WA, Delivery, Agent1, RA)}$$

$$\frac{Order(Agent1, RA, Agent3, WA) : WellKnownDebtor(Agent1)}{Prohibition(Agent3, WA, Delivery, Agent1, RA)}$$

4 Conflict Resolution

Various approaches for conflict resolution have been proposed in the last decade. It seems that the common ground for most of them is the ascription of priorities to norms [29, 18], policies [24, 21, 9], roles [6], based on some criterion, which may be domain dependent or independent. Belief revision [29], goal reduction and decision based on utility [19], and conflicting provision voidance [1] are some of the other proposed strategies for conflict resolution.

Conflict resolution in DfL may be performed using Brewka's [3] proposal that enables us to define and apply priorities on default rules dynamically.

Brewka defines a Preferential Default Theory (PDfT) as a triple (*W*, *D*, *name*), where name is a function that assigns names to default rules D. The extension of a PDfT is derived in the same way as in a DfT.

What makes PDfTs really useful is that the ascription of priorities to default rules may, itself, be done dynamically. Using dynamic priorities, we generate preferred extensions, each of which indicates a transaction plan. Priorities amongst ground defaults may be defined dynamically either by making different assumptions or by specifying domain-dependent criteria. The general pattern for ascribing priorities dynamically takes the form of a default rule:

$$\frac{Rule(d1, v1) \wedge Rule(d2, v2) \wedge criterion : assumptions}{d1 < d2}$$

Here $d1$, $d2$ are variables that denote names of ground defaults; $Rule(d1,v1)$ denotes a ground default $d1$ and its set of entities of interest v1. The intended interpretation of this rule is: if two defaults $d1$ and $d2$ apply and some criterion is satisfied between entities of interest, then $d1$ takes priority over $d2$, if certain assumptions may consistently be made.

Three general strategies for defining such criteria have been discussed in the literature, namely hierarchies of entities of interest, time and specificity of norms. Table 1 summarizes one possible way in which the patterns of normative conflicts that we discussed may be used by specific strategies. Given a particular normative conflict, different resolution strategies may be applied depending on our specific criterion of interest.

Table 1 Resolution Strategies for Normative Conflict Patterns

Strategy	Conflict Pattern	Criterion
Hierarchy	• Primitive normative conflicts • Inter-policy/inter-role conflicts • Conflict of duties/interests • Agreement-based conflicts • Type of action-based conflicts • Conflicts between assumptions and knowledge	e.g. prohibitions overrides all other normative notions, explicit knowledge overrules assumptions, obligations to reparatory actions should be met first, a regular client has priority
Temporality	• Time interval-based conflicts	e.g. the oldest obligation takes priority, or the shortest deadline takes priority
Specificity	• Exceptions	e.g. the most specific rule overrides all others

For instance, consider the case where two norms ($D1$ and $D2$) that define conflicting obligations for *Agent*3 are active. The first one is initiated at $ET1$ and it is towards retailer Agent1 who is a regular client. It sets an obligation to perform delivery until $IT1$. The second one is towards retailer *Agent*2, it is initiated at $ET2$ and defines a reparatory obligation to perform delivery until IT2. The relation between time points is as follows: $ET1 < ET2 < IT2 < IT1$. There is information that can be used to determine different conflict resolution criteria. The strategy of temporality based on external time may give priority to $D1$ as it was initiated first. On the other hand, temporality based on internal time may give priority to $D2$ since it has a shorter deadline. Another alternative, using the strategy of hierarchy is to give precedence to $D1$, because Agent1, as a regular client, takes precedence over *Agent*2. Or, we may give precedence to $D2$, because it concerns a reparatory action, if we choose to assign higher priority to secondary norms over primary ones. It should be clear that various combinations of these criteria may also be defined based on the agent's current knowledge and the assumptions it makes.

Finally, a fourth general strategy may also be applicable, by exploiting the fact that we employ DfL: an agent may ascribe priorities between default rules, based on the number of assumptions used. For instance a cautious agent may give priority to rules with fewer assumptions, while a risky agent may give priority to rules with more assumptions.

5 Conclusions and Future Work

The analysis, representation and management of normative conflicts has been the focus of much research in recent years, from a variety of perspectives.

Moffett *et al.* [24], Lupu *et al.* [21] and Dunlop *et al.* [8, 9] address conflicts from the Distributed Systems Management viewpoint by specifying policies as a way to determine and influence management behavior. We have shown how the basic types of conflict presented in these approaches may be seen as instances of our primitive patterns in DfL. Both [24] and [21] focus on the detection and resolution of syntactic conflicts at compile-time, by proposing static priority assignment. On the other hand, work in [8, 9], which addresses temporal reasoning about conflicts, concentrates on run-time conflict detection. Our approach is intended for conflict detection and resolution at run-time. Cholvy *et al.* in [5, 6] accept only inter-role conflicts and propose a solution that is based on the concept of role and regulation respectively. Contrary to this approach we accept intra-role conflicts and have shown how their conflict patterns map onto our primitive ones. None of the above approaches supports defeasible reasoning.

Broersen *et al.* in [4] deal with different kinds of conflicts: they are interested in conflicts arising between an agent's beliefs, obligations, intensions and desires. Although they, too, use DfL, they only use normal defaults, thus requiring agents to have complete knowledge. They do not address conflicts in a temporal setting. Other approaches, such as [13, 12, 25], that also support nonmonotonic reasoning with e-contracts do not presented in detail a discussion on the conflict patterns they consider and priorities are statically defined.

Abraham and Bacon in [1] examine normative conflicts but their focus corresponds to only a part of our set of primitive patterns. Although the absence of implicit knowledge is mentioned as a conflicting pattern (our pattern F), no resolution is proposed because no assumption-based reasoning is supported. Kowalski [19] is concerned with goal-driven conflict detection and resolution and attempts to unify logic with decision theory.

Our current work focuses on developing a computational tool based on Reiter's DfL and its major variations, that supports temporal defeasible reasoning as well as conflict detection and resolution as presented in this report and in [10, 11]. We also find Kowalski's proposal to use decision theory attractive as a means to derive preferred extensions.

Acknowledgments

This work was supported by the European Social Fund (75%) and the Greek Secretariat for Research and Technology (25%) through the 3rd Community Support Programme - Measure 8.3 (PENED2003-03EΔ466).

References

[1] Alan S. Abrahams and Jean M. Bacon. The life and times of identified, situated, and conflicting norms. In *DEON '02: Deontic Logic in Computer Science, 6th International Workshop on Deontic Logic in Computer Science*, 2002.

[2] Grigoris Antoniou. A tutorial on default logics. *ACM Computer Surveys*, 31(4):337–359, 1999.

[3] Gerhard Brewka. Reasoning about priorities in default logic. In *AAAI '94: Proceedings of the twelfth national conference on Artificial intelligence (vol. 2)*, pages 940–945, Menlo Park, CA, USA, 1994. American Association for Artificial Intelligence.

[4] Jan Broersen, Mehdi Dastani, Joris Hulstijn, Zisheng Huang, and Leendert der van Torre. The BOID architecture: conflicts between beliefs, obligations, intentions and desires. In *AGENTS '01: Proceedings of the Fifth International Conference on Autonomous Agents*, pages 9–16, Montreal, Canada, 2001. ACM Press.

[5] Laurence Cholvy and Frederic Cuppens. Solving normative conflicts by merging roles. In *ICAIL '95: Proceedings of the 5th international conference on Artificial intelligence and law*, pages 201–209. ACM Press, 1995.

[6] Laurence Cholvy and Frederic Cuppens. Reasoning about norms provided by conflicting regulations. In Henry Prakken and Paul McNamara, editors, *Norms, Logics and Information Systems. New Studies in Deontic Logic and Computer Science*, pages 247–264. IOS Press, Amsterdam, 1998.

[7] Aspassia Daskalopulu. Modeling legal contracts as processes. In *Legal Information Systems Applications, 11th International Workshop on Database and Expert Systems Applications (DEXA'00)*, pages 1074–1079. IEEE Computer Society, 2000.

[8] Nicole Dunlop, Jadwiga Indulska, and Kerry Raymond. Dynamic conflict detection in policy-based management systems. In *EDOC '02: Proceedings of the Sixth International Enterprise Distributed Oblect Computing Conference*, pages 15–26, Washington, DC, USA, 2002. IEEE Computer Society.

[9] Nicole Dunlop, Jadwiga Indulska, and Kerry Raymond. Methods for conflict resolution in policy-based management systems. In *EDOC '03: Proceedings of the 7th International Conference on Enterprise Distributed Object Computing*, pages 98–111, Washington, DC, USA, 2003. IEEE Computer Society.

[10] Georgios K. Giannikis and Aspassia Daskalopulu. Defeasible reasoning with e-contracts. In *IAT '06: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 690–694. IEEE Computer Society, 2006.

[11] Georgios K. Giannikis and Aspassia Daskalopulu. e-Contract representation for defeasible temporal reasoning and conflict management. Technical Report A004.678ΓΙΑ, Department of Computer and Communications Engineering, University of Thessaly, A004.678ΓΙΑ, 2006.

[12] Guido Governatori and Duy Hoang. A semantic web based architecture for e-contracts in defeasible logic. In *RuleML*, pages 145–159, 2005.

[13] Benjamin N. Grosof. Representing e-commerce rules via situated courteous logic programs in RuleML. *Electronic Commerce Research and Applications*, 3(1):2–20, 2004.

[14] H. Herrestad and C. Krogh. Deontic logic relativised to bearers and counterparties, 1995.

[15] John F. Horty. Moral dilemmas and nonmonotonic logic. *Journal of Philosophical Logic*, 23(1):35–65, 1994.

[16] John F. Horty. Reasoning with moral conflicts. *Noûs*, 37(4):557–605, 2003.

[17] Andrew J.I. Jones and Marek J. Sergot. A formal characterisation of institutionalised power. *Journal of the IGPL*, 4(3):427–443, 1996.

[18] Martin J. Kollingbaum and Timothy J. Norman. Strategies for resolving norm conflict in practical reasoning. In *ECAI Workshop CEAS*, 2004.

[19] Robert A. Kowalski. A logic-based model for conflict resolution, 2003. http://www.doc.ic.ac.uk/ rak/.

[20] Ronald M. Lee. Bureaucracies as deontic systems. *ACM Trans. Inf. Syst.*, 6(2):87–108, 1988.

[21] Emil Lupu and Morris Sloman. Conflicts in policy-based distributed systems management. *IEEE Trans. Software Eng.*, 25(6):852–869, 1999.

[22] David Makinson. On the formal representation of rights relations. *Journal of Philosophical Logic*, 15(4):403–425, 1986.

[23] Rafaél Hernández Marín and Giovanni Sartor. Time and norms: a formalisation in the event-calculus. In *ICAIL '99: Proceedings of the 7th international conference on Artificial intelligence and law*, pages 90–99, New York, NY, USA, 1999. ACM Press.

[24] Jonathan D. Moffett and Morris S. Sloman. Policy conflict analysis in distributed system management. *Journal of Organizational Computing*, 1993.

[25] Adrian Paschke, Martin Bichler, and Jens Dietrich. ContractLog: An approach to rule based monitoring and execution of service level agreements. In *RuleML*, pages 209–217, 2005.

[26] Henry Prakken and Marek J. Sergot. Contrary-to-duty obligations. *Studia Logica*, 57(1):91–115, 1996.

[27] Raymond Reiter. A logic for default reasoning. *Artif. Intell.*, 13(1-2):81–132, 1980.

[28] Lamber M. M. Royakkers. *Extending Deontic Logic for the Formalisation of Legal Rules*. Kluwer Academic Publishers, 1998.

[29] Giovanni Sartor. Normative conflicts in legal reasoning. *Artificial Intelligence and Law*, 1(2-3):209–235, 1992.