Identifying Indirect Normative Conflicts using the WordNet Database

Jéssica Soares dos Santos[1] and Viviane Torres da Silva[2]
[1]*Computer Science Department, Universidade Federal Fluminense, Niterói, Brazil*
[2]*IBM Research (on leave from Universidade Federal Fluminense), Rio de Janeiro, Brazil*

Keywords: Multi-agent Systems, Normative Conflicts, Conflict Detection, WordNet.

Abstract: A challenging issue in Multi-agent systems governed by multiple norms is to deal with normative conflicts, which are situations where the fulfilment of a norm violates another one. There are several approaches in the literature to detect conflicting norms. Some kinds of conflicts, here called direct conflicts, can be detected through a simple comparison between the elements of the norms in order to check if they apply to the same elements. For instance, if an obligation and a prohibition are applied to the *same* entity and govern the *same* behavior in periods of time that intersects, these norms are in conflict. However, there are conflicts, here called indirect conflicts, that can only be detected if the relationships among the elements of the norms are taken into account. The majority of approaches that are able to detect indirect conflict considers the relationships of the application domain that have been previously defined by the designer. Different from those approaches, this paper focuses on the detection of indirect conflicts by taking into account domain-independent relationships that have not been declared by the designer of the multi-agent system. Our proposal searches for domain-independent relationships among the elements of the norms in the *WordNet* database.

1 INTRODUCTION

In open Multi-Agent Systems (MASs) norms have been applied as a means to restrict and guide the behavior of autonomous and independently designed software agents in order to avoid undesirable behaviour. A norm is commonly associated with a deontic concept (obligation, permission or prohibition) and defines which actions an agent can or cannot perform (Grossi et al., 2010). Due to the numeral norms that may be necessary to govern a normative MAS, conflicts between the norms may arise. A conflict between two norms occurs when the fulfilment of a norm violates the other norm. For this reason, to guarantee the proper functioning of a normative MAS, mechanisms for dealing with normative conflicts are needed. A normative conflict can be classified as:

- **Direct Conflict:** it involves two norms that are associated with the same addressee, regulate the same behavior and have opposite or contradictory deontic modalities (obligation *versus* prohibition or permission *versus* prohibition). This kind of conflict can be detected through a simple analysis of the norm elements (i.e., the addressee, behaviour and context of the norm). Thus, a direct normative conflict arises, for instance, when a norm obliges an agent to perform an action in a given organization and another norm prohibits the same addressee to execute the same action in a same organization.
- **Indirect Conflict:** it involves two norms whose elements are not the same but are related. It can only be detected when relationships among elements of the norms are identified. The deontic concepts associated with the norms involved in an indirect conflict can be opposite, contradictory or equal. For instance, an indirect normative conflict arises between two norms when both norms are addressed to the same agent and one is prohibiting the execution of an action and the other is obligating the execution of another action that is a specialization of this action. Since the more general action are being prohibited the more concrete action cannot be executed.

The aim of our research is to develop a mechanism able to detect indirect normative conflicts by using the WordNet database (Miller, 1995) to identify the domain-independent relationships among the elements of the norms. WordNet is a public lexical database that stores relationships among words.

The main difference between our approach and others also able to detect indirect conflicts is that our

Santos, J. and Silva, V.
Identifying Indirect Normative Conflicts using the WordNet Database.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 2*, pages 186-193
ISBN: 978-989-758-187-8

approach detects domain-independent relationships. Other approaches, such as (da Silva and Zahn, 2014; Aphale et al., 2012; Sensoy et al., 2012), focus on the identification of domain-dependent relationships that have been defined by the application designer. In fact, our approach and the ones published in the literature are complementary since the identification of all kinds of indirect conflicts can only be detected if both kinds of relationships are considered.

This paper is organized as follows. Section 2 presents related work. Section 3 describes the kinds of relations that are defined in the WordNet database. Section 4 presents the norm definition adopted in this paper. Section 5 describes the kinds of relationships explored by our approach and formalizes the rules for conflict detection. Section 6 presents the detection algorithm and describes our mechanism. Finally, Section 7 concludes and presents some future work.

2 RELATED WORK

Several techniques have been proposed to detect normative conflicts. Some of them only are able to detect direct conflicts (Li et al., 2014; Vasconcelos et al., 2012; dos Santos Neto et al., 2013; Gunay and Yolum, 2013) while others can also detect indirect conflicts (da Silva and Zahn, 2014; Aphale et al., 2012; Sensoy et al., 2012; Fenech et al., 2009; Giannikis and Daskalopulu, 2011). Each approach presents a different definition for norms and can detect different kinds of normative conflicts. The majority of approaches to detect indirect conflicts only deals with relationships among actions, i.e., they do not consider relationships between other elements of the norm.

Since our approach focuses on the detection of indirect conflicts, we will present a brief description about the kinds of relationships that related approaches consider in order to detect indirect normative conflicts. Note that none of the approaches are able to detect domain-independent relationships. All of them assume that the domain-dependent relationship are provided to the conflict checker in order to check for conflicts.

2.1 Action Relationships

This subsection describes the action relationships that have been found in the literature.

• **Refinement:** it defines a relation of specialization among actions, i.e., it relates a sub action to a super action. It is described in the following approaches (da Silva and Zahn, 2014; Zahn, 2015);
• **Composition:** it defines a relation of composition among actions, i.e., it determines that an action is composed of another one. It is described in (da Silva and Zahn, 2014; Zahn, 2015; Vasconcelos et al., 2009; Aphale et al., 2012; Sensoy et al., 2012);
• **Orthogonality:** it defines actions that cannot be performed simultaneously by the same entity, as presented in (da Silva and Zahn, 2014; Zahn, 2015; Fenech et al., 2008, 2009; Giannikis and Daskalopulu, 2011, 2009);
• **Dependency:** it defines actions that are preconditions of other actions. It is described in the following approaches (da Silva and Zahn, 2014; Zahn, 2015; Aphale et al., 2012; Sensoy et al., 2012);
• **Actions' Side Effects:** it determines the side-effects of the execution of an action, as detailed in (Aphale et al., 2012; Sensoy et al., 2012; Kollingbaum and Norman, 2004; Kollingbaum et al., 2006; Kollingbaum and Norman, 2006; Kollingbaum et al., 2007).

Additionally, the approach presented in (Vasconcelos et al., 2009) also can detect normative conflicts that occur due to the delegation of tasks among agents. Moreover, the work in (Zahn, 2015) is able to analyze relationships (such as refinement, dependency, orthogonality and composition) between: (i) two actions; (ii) two states; or (iii) an action and a state.

2.2 Entity Relationships

This subsection describes the entity relationships that have been found in the literature.

• **Play:** it relates an agent to the roles it can play. It is described in the following approaches (da Silva and Zahn, 2014; Zahn, 2015; Cholvy and Cuppens, 1995, 1998);
• **Play-in:** it relates an agent to the organization where it is playing a role, as presented in (da Silva and Zahn, 2014;);
• **Ownership:** it relates roles to the organizations where they were defined and can be played, as detailed in (da Silva and Zahn, 2014; Zahn, 2015);
• **Hierarchy:** it relates a sub role to a super role. It is defined in (da Silva and Zahn, 2014; Zahn, 2015).

2.3 Context Relationships

This subsection describes the context relationships that have been found in the literature. The context of a norm represents the scope where it is defined, that is, where the norm must be applicable. For instance, a norm can be applied to an environment or to a given organization.

• **Inhabit:** it relates an entity to the environment that is its habitat. It is described in the following approaches (da Silva and Zahn, 2014; Zahn, 2015);
• **Hierarchy:** it relates a sub context to its super context, as presented in (da Silva and Zahn, 2014; Zahn, 2015).

3 WORDNET

The WordNet is a lexical database that stores relationships among words, which can be nouns, verbs, adjectives and adverbs. WordNet groups words that share the same meaning in a given context. Each group of synonymous words is called *synset*. A *synset* is the building block of WordNet and all words in a *synset* denote the same concept. Each *synset* has a brief description (called "gloss") of its meaning and may be associated with short sentences that exemplify the use of synset members. For instance, the nouns "error" and "mistake" are grouped in a same synset whose *gloss* is "part of a statement that is not correct". Synsets are interlinked through semantical and lexical relations. We will use WordNet to identify relationships among the elements of the norms that have not been defined in the domain application. The relations that are stored in WordNet are described in the next subsection.

3.1 WordNet Relations

WordNet defines six relationships, as follows:

• **Synonymy:** it is the main relation of WordNet. Synonymous are words with the same meaning in a given context. As stated in the beginning of this Section, in WordNet words are grouped into synsets (synonymous sets) and all words of a synset denote the same concept. A word may appear in more than one synset if it is associated with different meanings. This relation is defined to nouns, verbs, adjectives and adverbs.
• **Hyponymy/Hypernymy:** it relates a subset to a superset since it links one synset more general to another one more specific. For instance, *whale* is a hyponym of *mammal* because *whale* is a kind of *mammal*. Similarly, *mammal* is a hypernym of *whale*. This relationship is similar to hierarchy described in Section 2. This relation is defined to nouns.
• **Meronymy/Holonymy:** it corresponds to the relation part-whole. For instance, *seat* is a meronym of *chair* because *seat* is part of *chair*. Similarly, *chair* is a holonym of *seat*. This relation is only defined to nouns.
• **Entailment:** it defines that a verb entails another one. This relation is unilateral. For instance, *to buy* entails *to pay* because *to buy* we need *to pay*. Then, if two verbs are related by an entailment relation and the event denoted by the first verb occur, the action denoted by the second verb will also occur. This relation is only defined to verbs.
• **Antonymy:** it relates a word to another one with opposite meaning. For instance, *clean* is antonym of *dirty* and vice-versa. This relation can be regarded a special case of the orthogonality cited in Section 2. This relation is defined to nouns, verbs and adverbs.
• **Troponymy/Hypernymy:** it relates a verb denoting an action to another one denoting a manner of doing this action. For instance, *to lisp* is a manner of *to talk*. Thus, *to lisp* is a troponym of *to talk*. This relationship is similar to refinement described in Section 2. This relation is only defined to verbs.

4 NORM DEFINITION

The norm definition used in this paper is based on the definition presented in (Zahn, 2015). We consider that norms oblige, prohibit or permit an entity to perform an action in a given context during a period of time. Additionally, we consider that agents inhabit environments and play roles in organizations.

Our definition of norm is more expressive than the one described in (Zahn, 2015) because we extend it including an optional field "obj", which corresponds to a parameter of the action being regulated by the norm that describes an object applied to the action. In order to exemply, we can consider the following actions: dress(skirt) and eat(banana).

A norm is a tuple of the form:

Norm = ⟨id, d, c, e, act (obj), ac, dc⟩,

where *id* is the norm identifier, *d* is a deontic concept from the set {obligation, permission, prohibition}; $c \in C$ is the context where the norm is defined (i.e., an organization $org \in O$ or an environment $env \in Env$.); $e \in E$ is the entity whose action is being regulated by the norm (i.e., an agent $a \in A$, an organization $org \in O$ or a role $r \in R$. We use the symbol "_" to determine that a norm is addressed to all entities of a given context); *act* ∈

Act is the action being regulated; *obj* ∈ Obj is an object associated with the action (it is an optional field. We will use the symbol "_" to represent that an action is not associated with any object.); *ac* ∈ Cnd is the condition that activates the norm; *dc* ∈ Cnd is the condition that deactivates the norm. These conditions are dates represented in the format: MM/dd/yyyy HH:mm:ss.

5 KINDS OF RELATIONSHIPS

In this section we present the relationships that our proposal will analyze in order to detect indirect normative conflicts. First, the elements of a norm are mapped to nouns or verbs and then we identify the relationships among such elements.

We adopt the norm propagation method described in (Zahn, 2015; da Silva et al., 2015), which determines propagation rules to create new norms in order to facilitate the detection process. Propagation rules propagate norms according to the relationships between entities or contexts. After the norm propagation, the detection method only need to verify, for each pair of norms addressed to the same entity and context, if the actions regulated by the norms are related. To avoid the unnecessary propagation of norms addressed to synonymous entities/contexts, we identify and locally rewrite synonymous entities/contexts before propagating norms. We use First-order logic (Smullyan, 1995) to define norm propagation constraints. For the sake of simplicity, we omit the norm identifier *id*.

Since our norm definition (Section 4) considers that an action may be applied over an object, norms associated with related actions only can conflict with each other if both actions are associated with the equivalent objects (i.e., synonyms or equal) or they are not associated with objects.

The identification of relationships and the propagations do only consider the entities, contexts, and actions defined in the set of norms being considered.

5.1 Context Relationships

In this subsection we describe the context relationships that can be captured by our approach. All contexts are mapped to nouns and the relationships defined in the WordNet that applied to nouns are investigated. We do not consider the antonymy relationship because there is not a case where a norm must be propagated due to an antonymy relation between contexts.

- **Synonymy Among Contexts:** we use the WordNet relation "Synonymy" to identify synonymous contexts. For instance, the context *USA* is equivalent to the context *United States of America*.

  Rewriting Rule: Synonymous contexts are locally rewritten in order to unify the norm elements that are equivalent. Thus, we iterate over the set of contexts used in the set of norms and create a list where each item is a pair of synonyms contexts. After that, we replace each context for its synonym stored in the synonyms list. For instance, if there is a norm n1 that is associated with the context *USA* and there is a norm n2 associated with the context *United States of America*, since both contexts are synonyms, norm n2 is locally rewritten to be associated with the context *USA*.

- **Specialization Among Contexts:** we use the WordNet relation "Hyponymy/ Hypernymy" to identify related contexts. For instance, the context *Brazil* is a hyponym of the context *South America*.

  Propagation Rule: If a norm is addressed to an organization and does not specify a specific entity, so it must be addressed to all its sub organizations.

$$\exists o_1 (\langle d, o_1, \_, act(obj), ac, dc\rangle \in Norm \rightarrow \forall o_2 (HyponymyHypernymy(o_2, o_1) \rightarrow \langle d, o_1, o_2, act(obj), ac, dc\rangle \in Norm))$$

where d ∈ {obligation, permission, prohibition}; o_1, o_2 ∈ O; act ∈ Act; obj ∈ Obj; ac ∈ Cnd; dc ∈ Cnd.

- **Part-whole Among Contexts:** we use the WordNet relation "Meronymy/ Holonymy" to identify related contexts. For instance, the context *intensive care unit* (ICU) is part of the context *hospital*.

  Propagation Rule: If a norm is addressed to an organization and does not specify a specific entity, it must be addressed to all organizations that are part of the given organization.

$$\exists o_1(\langle d, o_1, \_, act, ac, dc\rangle \in Norm \rightarrow \forall o_2 (MeronymyHolonymy(o_2, o_1) \rightarrow \langle d, o_1, o_2, act, ac, dc\rangle \in Norm))$$

where d ∈ {obligation, permission, prohibition}; o_1, o_2 ∈ O; act ∈ Act; obj ∈ Obj; ac ∈ Cnd; dc ∈ Cnd.

5.2 Entity Relationships

In this subsection we describe the entity relationships that can be captured by our approach. All entities are mapped to nouns and the relationships defined in the WordNet that applied to nouns are investigated. We do not consider the part-whole relationship among entities because we did not find cases where an entity is composed of parts. Our approach can be easily extended to support the relationship part-whole among entities if necessary. We also do not consider the relationship antonymy among entities because it does not imply that a norm must be propagated.

• **Synonymy Among Entities:** we use the WordNet relation "Synonymy" to identify synonymous entities. For instance, the entity *child* is equivalent to the entity *kid*.

Rewriting Rule: Synonymous entities are locally rewritten in order to unify the norm elements that are equivalent. This process is similar to the one described in the "synonymy among contexts", in Section 5.1.

• **Specialization Among Entities:** we use the WordNet relation "Hyponymy/ Hypernymy" to identify related entities. For instance, the role *angiologist* is hyponym of the role *doctor*.

Propagation Rule: If a norm is addressed to a hypernym entity (super role), then it must apply to all sub roles of the given super role. Only the roles defined in the domain ontology are compared.

$\exists r_1 (\langle d, c, r_1, act\ (obj), ac, dc\rangle \in Norm \rightarrow$
$\forall r_2\ (HyponymyHypernymy\ (r_2, r_1) \rightarrow$
$\langle d, c, r_2, act\ (obj), ac, dc\rangle \in Norm))$

where d ∈ {obligation, permission, prohibition}; $c \in C$; r_1, $r_2 \in R$; act ∈ Act; obj ∈ Obj; ac ∈ Cnd; dc ∈ Cnd.

5.3 Action Relationships

In this subsection we describe the action relationships that can be captured by our approach. All actions are mapped to verbs and the relationships defined in the WordNet that applied to verbs are investigated.

• **Synonymy Among Actions:** we use the WordNet relation "Synonymy" to identify that an action is equivalent to another one. For instance, the actions *to cooperate* and *to collaborate* are synonyms.

Conflict Rule: Norms associated with synonymous actions are in conflict if one is a prohibition and the other is a permission or obligation, and both actions are applied to the equivalent objects (synonym or equal) or they are not associated with objects.

$\exists act_1\ \exists act_2\ ((\langle prohibition, c, e, act_1\ (obj_1), ac_1, dc_1\rangle \in Norm$
$\wedge \langle d, c, e, act_2\ (obj_1), ac_2, dc_2\rangle \in Norm$
$\wedge (Synonymy\ (act_1, act_2)) \rightarrow$
$conflict\ (\langle prohibition, c, e, act_1\ (obj_1), ac_1, dc_1\rangle,$
$\langle d, c, e, act_2\ (obj_1), ac_2, dc_2\rangle))$

where d ∈ {obligation, permission}; $c \in C$; $e \in E$; act_1, $act_2 \in Act$; $obj_1 \in Obj$; ac_1, $ac_2 \in Cnd$; dc_1, $dc_2 \in Cnd$; $dc_2 \geq ac_1$; and $dc_1 \geq ac_2$.

• **Specialization Among Actions:** we use the WordNet relation "Troponymy/Hypernymy" to identify related actions. For instance, the action *to talk* is a troponym of the action *to communicate*.

Conflict Rule: Norms associated with specialization actions are in conflict if the super action (hypernym) is prohibited and the sub action (troponym) is permitted or obliged, and both actions are applied to equivalent objects (synonym or equal) or they are not associated with objects.

$\exists act_1\ \exists act_2\ ((\langle prohibition, c, e, act_1\ (obj_1), ac_1, dc_1\rangle \in Norm$
$\wedge \langle d, c, e, act_2\ (obj_1), ac_2, dc_2\rangle \in Norm$
$\wedge TroponymyHypernymy\ (act_2, act_1)) \rightarrow$
$conflict\ (\langle prohibition, c, e, act_1\ (obj_1), ac_1, dc_1\rangle,$
$\langle d, c, e, act_2\ (obj_1), ac_2, dc_2\rangle))$

where d ∈ {obligation, permission}; $c \in C$; $e \in E$; act_1, $act_2 \in Act$; $obj_1 \in Obj$; ac_1, $ac_2 \in Cnd$; dc_1, $dc_2 \in Cnd$; $dc_2 \geq ac_1$; and $dc_1 \geq ac_2$.

• **Entailment:** we use the WordNet relation "Entailment" to identify related actions. For instance, the action *to buy* entails the action *to pay*.

Conflict Rule: Norms associated with entailment actions are in conflict if the entailed action is a prohibition and the other action is an obligation or permission, and both actions are applied to equivalent objects (synonym or equal) or they are not associated with objects.

$\exists act_1\ \exists act_2\ ((\langle d, c, e, act_1\ (obj_1), ac_1, dc_1\rangle \in Norm$
$\wedge \langle prohibition, c, e, act_2\ (obj_1), ac_2, dc_2\rangle \in Norm$
$\wedge Entailment\ (act_1, act_2)) \rightarrow$
$conflict\ (\langle d, c, e, act_1\ (obj_1), ac_1, dc_1\rangle,$
$\langle prohibition, c, e, b_2\ (obj_1), ac_2, dc_2\rangle))$

where d ∈ {obligation, permission}; $c \in C$; $e \in E$; act_1, $act_2 \in Act$; $obj_1 \in Obj$; ac_1, $ac_2 \in Cnd$; dc_1, $dc_2 \in Cnd$; $dc_2 \geq ac_1$; and $dc_1 \geq ac_2$.

• **Antonymy Among Actions:** we use the WordNet relation "Antonymy" to identify that an action is opposite to another one. For instance, the action *to enter* is antonym of the action *to leave*.

Conflict Rule: Norms associated with antonym actions are in conflict if they are obligations, and both actions are applied to equivalent objects (synonym or equal) or they are not associated with objects. Note that a conflict does not occurs if the norms are permissions or are prohibitions.

$\exists b_1\ \exists b_2\ ((\langle obligation, c, e, act_1\ (obj_1), ac_1, dc_1\rangle \in Norm$
$\wedge \langle obligation, c, e, act_2\ (obj_1), ac_2, dc_2\rangle \in Norm$
$\wedge Antonymy\ (act_2, act_1)) \rightarrow$
$conflict\ (\langle obligation, c, e, act_1\ (obj_1), ac_1, dc_1\rangle,$
$\langle obligation, c, e, act_2\ (obj_1), ac_2, dc_2\rangle))$

where $c \in C$; $e \in E$; act_1, $act_2 \in Act$; $obj_1 \in Obj$; ac_1, $ac_2 \in Cnd$; dc_1, $dc_2 \in Cnd$; $dc_2 \geq ac_1$; and $dc_1 \geq ac_2$.

6 ALGORITHM

In this section we present pseudocodes and describe our detection method. The WordNet Conflict Checker[1] checkes for conflicts between a set of norms by considering the relationships between their elements (entities, contexts, actions) that are defined in the WordNet database. The conflict checker was implemented by using Java language. To execute searches over the WordNet database, we use JWNL (Walenz and Didion, 2011), a Java library that allow us to connect the information from an offline WordNet database to a Java program. The main steps of our detection method (see Algorithm 1) are described as follows. Due to the limitation of space we omit the algorithms of identification of synonyms and norm propagation.

- **Identify Synonymous:** We use the WordNet database (synonymy relationship) to identify the synonymous contexts and entities and locally rewrite the norms to unify the syntax of the norms and avoid the propagation of equivalent norms. In this step we store the pairs of synonymous entities/contexts in a list. Each item of the list is a pair of the form (key, value), where the key is the entity to be replaced for the entity stored in the value. For instance, if the domain ontology has the entities: child, kid, minor, the algorithm will iterate sequentially through the entities and create the list: {(kid, child), (minor, child)}. Then, norms addressed to the entity kid or to the entity minor will be locally rewritten to be addressed to the entity child.
- **Norm Propagation:** The second step of the algorithm is to propagate norms from more general contexts (or entities) to more specific contexts (or entities). This process uses the WordNet information in order to identify the relationships hierarchy and part-whole between contexts and entities. In this step new norms are created according to the relations identified. This step is described in Section 5.1 and 5.2.
- **Get related Norms:** In the third step, the detection mechanism gets each pair of norms and verifies whether the norms are addressed to the same entity, to the same context and if their period of validity (activation and deactivation conditions) intersect. This step is described in Algorithm 1. After that, the mechanism verifies if there is a direct conflict. Otherwise, the mechanism uses the WordNet database to verify if there is an indirect conflict (see Algorithm 2).
- **WordNet Conflict Rules:** In the fourth step, the conflict rules of the WordNet relationships (synonymy, specialization, antonymy, entailment) are applied in order to detect conflicts. These rules were specified in Section 5. The algorithm analyzes the deontic concept of the norms and searches for relationships among the actions or objects using WordNet database. The algorithms 3, 4, 5, 6 detail the application of conflict rules. In our implementation available to download, the program informs to the user whether each pair of compared norms are in conflict or not and presents the reason of the conflict. For the sake of simplicity, Algorithm 1 only informs to the user whether there is a normative conflict or not.

So, in summary, the conflict checker algorithm uses the WordNet database in order to: (i) unify the syntax of the elements defined in the set of norms; (ii) propagate norms addressed to more general entities (or contexts) to more specific entities (or contexts); and (iii) search relationships between the actions defined in the set of norms to detect possible indirect normative conflicts.

Algorithm 1: Check Conflicts.

```
Require: N: set of norms, E: set of entities, C: set of contexts,
W: WordNet database
  function EXECUTE (N, E, C, W)
     conflictTime ← false
     conflictAction ← false
     synonymEntities ← GETSYNONYMENTITIES (E, W)
     synonymContexts ← GETSYNONYMCONTEXTS (C, W)
     N ← PROPAGATEENTITY (N, synonymEntities, E, W)
     N ← PROPAGATECONTEXT (N, synonymContexts,
          synonymEntities, E, W)
     for all n1 ∈ N do
         for all n2 ∈ N do
             conflictTime ← timeIntersect (n1, n2)
               if ((conflictTime = true) ∧ (n1.e = n2.e) ∧
                  (n1.c = n2.c)) then
                  conflictAction ← CHECKACTION (n1, n2, W)
    return conflictAction
```

Algorithm 2: Check Action.

```
Require: n1, n2: norms, W: WordNet database
  function CHECKACTION (n1, n2, W)
     if ((n1.act = n2.act) ∧ (n1.obj = n2.obj) ∧
        (n1.d = "FORBIDDEN") ∧
        (n2.d = ("OBLIGED" ∨ "PERMITTED"))) then
         return true
     if (SYNONYMYCR (n1, n2, W) ∨
        ANTONYMYCR (n1, n2, W) ∨
        SPECIALIZATIONCR (n1, n2, W) ∨
        ENTAILMENTCR (n1, n2, W) then
          return true
     return false
```

[1] Available at https://goo.gl/KeF5M3

Algorithm 3: Synonymy Conflict Rule.

```
Require: n1, n2: norms, W: WordNet database
  function SYNONYMYCR (n1, n2, W)
    if (SYNONYMY (n1.act, n2.act, W) ∧
      ((n1.obj = n2.obj) ∨ SYNONYMY (n1.obj, n2.obj, W)) ∧
       (n1.d = "FORBIDDEN") ∧
       (n2.d = ("OBLIGED" ∨ "PERMITTED"))) then
       return true
    return false
```

Algorithm 4: Antonymy Conflict Rule.

```
Require: n1, n2: norms, W: WordNet database
  function ANTONYMYCR (n1, n2, W)
    if (ANTONYMY (n1.act, n2.act, W) ∧
      ((n1.obj = n2.obj) ∨ SYNONYMY (n1.obj, n2.obj, W)) ∧
       (n1.d = "OBLIGED") ∧ (n2.d = "OBLIGED")) then
       return true
    return false
```

Algorithm 5: Specialization Conflict Rule.

```
Require: n1, n2: norms, W: WordNet database
  function SPECIALIZATIONCR (n1, n2, W)
    if (TROPONYMHYPERNYM (n2.act, n1.act, W) ∧
      ((n1.obj = n2.obj) ∨ SYNONYMY (n1.obj, n2.obj, W)) ∧
       (n1.d = "FORBIDDEN") ∧
       (n2.d = ("OBLIGED" ∨ "PERMITTED"))) then
       return true
    return false
```

Algorithm 6: Entailment Conflict Rule.

```
Require: n1, n2: norms, W: WordNet database
  function ENTAILMENTCR (n1, n2, W)
    if (ENTAILMENT (n1.act, n2.act, W) ∧
      ((n1.obj = n2.obj) ∨ SYNONYMY (n1.obj, n2.obj, W)) ∧
       (n2.d = "FORBIDDEN") ∧
       (n1.d = ("OBLIGED" ∨ "PERMITTED"))) then
       return true
    return false
```

7 CONCLUSIONS

The detection of normative conflicts is an essential key in a MAS governed by multiple norms. In the literature, there are several approaches that deal with conflicts among norms. The majority is only able to detect direct conflicts, but other ones can also detect indirect normative conflicts. The kinds of relationships that can be identified are different according to each approach. For the best of our knowledge, all approaches described in the literature only can detect indirect normative conflicts when the designer explicitly defines relationships among entities, contexts, actions and states. However, there are relationships that are independent of the domain. Our research focuses on the detection of indirect normative conflicts using relationships already defined in WordNet database to identify relationships among elements of the norms. We formalize the relationships explored and describe rules to detect normative conflicts. Our approach can detect relationships among the elements of the norms without analyzing relationships of the domain application. In order to improve our mechanism and make it more complete, we are studying the possibility of extending our approach to capture others relationships. We are in process of integrating detection methods that analyze domain application relationships.

REFERENCES

Aphale, M. S., Norman, T. J., and Sensoy, M. (2012). Goal directed conflict resolution and policy refinement. In *14th International Workshop on Coordination, Organizations*, Institutions and Norms in *Agent Systems*, Valencia, Spain. (COIN@AAMAS 2012).

Cholvy, L. and Cuppens, F. (1995). Solving normative conflicts by merging roles. In Proceedings of the 5th international conference on *Artifitial intelligence and law*, pages 201–209. ACM.

Cholvy, L. and Cuppens, F. (1998). Reasoning about norms provided by conflicting regulations. Norms, logics and information systems: new studies in *deontic logic and computer science,* pages 247–264.

da Silva, V. and Zahn, J. (2014). Normative conflicts that depend on the domain. In Balke, T., Dignum, F., van Riemsdijk, M. B., and Chopra, A. K., editors, Coordination, Organizations, Institutions, and Norms in Agent Systems IX, volume 8386 of Lecture Notes in Computer Science, pages 311–326. *Springer International Publishing*.

da Silva, V., Braga, C., and Zahn, J. (2015). Indirect normative conflict - conflict that depends on the application domain. In ICEIS 2015 - *Proceedings of the 17th International Conference on Enterprise Information Systems*, Volume 1, Barcelona, Spain, 27-30 April, 2015, pages 452–461.

dos Santos Neto, B., da Silva, V., and de Lucena, C. (2012). An architectural model for autonomous normative agents. In Barros, L., Finger, M., Pozo, A., Gimennez-Lugo, G., and Castilho, M., editors, Advances in Artificial Intelligence - SBIA 2012, Lecture Notes in *Computer Science*, pages 152–161. Springer Berlin Heidelberg.

dos Santos Neto, B., da Silva, V., and de Lucena, C. (2013). Developing goal-oriented normative agents: The nbdi architecture. In Filipe, J. and Fred, A., editors, Agents and Artificial Intelligence, volume 271 of Communications in Computer and Information Science, pages 176–191. *Springer Berlin Heidelberg*.

Fenech, S., Pace, G., and Schneider, G. (2009). Automatic conflict detection on contracts. In Leucker, M. and Morgan, C., editors, Theoretical Aspects of Computing - ICTAC 2009, volume 5684 of Lecture Notes in Computer Science, pages 200–214. *Springer Berlin Heidelberg*.

Fenech, S., Pace, G. J., and Schneider, G. (2008). Detection of conflicts in electronic contracts. *NWPT 2008*, page 34.

Gaertner, D., Garcia-Camino, A., Noriega, P., Rodriguez-Aguilar, J.-A., and Vasconcelos, W. (2007). Distributed norm management in regulated multiagent systems. In Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems, page 90. *ACM*.

Giannikis, G. K. and Daskalopulu, A. (2009). Normative conflicts-patterns, detection and resolution. In *WEBIST*, pages 527–532.

Giannikis, G. K. and Daskalopulu, A. (2011). Normative conflicts in electronic contracts. *Electronic Commerce Research and Applications*, 10(2):247–267.

Grossi, D., Gabbay, D., and van der Torre, L. (2010). The norm implementation problem in normative multi-agent systems. In Dastani, M., Hindriks, K. V., and Meyer, J.-J. C., editors, Specification and Verification of Multiagent Systems, pages 195–224. *Springer US*.

Gunay, A. and Yolum, P. (2013). Engineering conflict-free multiagent systems. In *First International Workshop on Engineering Multiagent Systems (EMAS)*.

Kollingbaum, M. and Norman, T. (2004). Strategies for resolving norm conflict in practical reasoning. In ECAI Workshop Coordination in *Emergent Agent Societies, volume 2004*.

Kollingbaum, M. and Norman, T. (2006). Informed deliberation during norm-governed practical reasoning. In Boissier, O., Padget, J., Dignum, V., Lindemann, G., E. Matson, S. Ossowski, J. Sichman, and J. Vázquez-Salceda, editors, Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems, volume 3913 of Lecture Notes in Computer Science, pages 183–197. *Springer Berlin Heidelberg*, 2006.

Kollingbaum, M., Norman, T., Preece, A., and Sleeman, D. (2006). Norm refinement: Informing the re-negotiation of contracts. In *ECAI 2006 Workshop on Coordination, Organization, Institutions and Norms in Agent Systems*, COIN@ ECAI, volume 2006, pages 46–51.

Kollingbaum, M., Norman, T., Preece, A., and Sleeman, D. (2007). Norm conflicts and inconsistencies in virtual organisations. In Noriega, P., Vázquez-Salceda, J., Boella, G., Boissier, O., Dignum, V., Fornara, N., and Matson, E., editors, Coordination, Organizations, Institutions, and Norms in Agent Systems II, volume 4386 of Lecture Notes in Computer Science, pages 245–258. *Springer Berlin Heidelberg*.

Li, T. (2013). Normative conflict detection and resolution in cooperating institutions. In *Proceedings of the TwentyThird international joint conference on Artificial Intelligence*, pages 3231–3232. AAAI Press.

Li, T., Jiang, J., Aldewereld, H., De Vos, M., Dignum, V., and Padget, J. (2014). Contextualized institutions in virtual organizations. In Balke, T., Dignum, F., van Riemsdijk, M. B., and Chopra, A. K., editors, Coordination, Organizations, Institutions, and Norms in Agent Systems IX, volume 8386 of Lecture Notes in Computer Science, pages 136–154. *Springer International Publishing*.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Sensoy, M., Norman, T., Vasconcelos, W., and Sycara, K. (2012). OWL-POLAR: A framework for semantic policy representation and reasoning, volume 12. *Elsevier*.

Smullyan, R. M. (1995). First-order logic. Courier Corporation.

Vasconcelos, W., Kollingbaum, M., and Norman, T. (2009). Normative conflict resolution in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 19(2):124–152.

Vasconcelos, W., García-Camino, A., Gaertner, D., Rodríguez-Aguilar, J. A., and Noriega, P. (2012). Distributed norm management for multi-agent systems. *Expert Systems with Applications*, 39(5):5990–5999.

Vasconcelos, W. and Norman, T. (2009). Contract formation through preemptive normative conflict resolution. Leibniz-Zentrum fur Informatik.

Walenz, B. and Didion, J. (2011). *Jwnl: Java wordnet library*.

Zahn, J. (2015). Um Mecanismo de Verificação de Conflitos Normativos Indiretos. Master's thesis, Instituto de Computação - Universidade Federal Fluminense (IC/UFF), Niterói, Brasil.