Modeling and Detecting Norm Conflicts in Regulated Organizations

Jie Jiang and Huib Aldewereld

Delft University of Technology, the Netherlands
{jie.jiang, h.m.aldewereld}@tudelft.nl

Abstract. In regulated organizations, norms may come from various regulation sources imposed by different institutions. With possibly conflicting values and interests, inconsistencies are likely to occur among these norms, e.g., one norm obliges some actions to be done while another norm prohibits the same actions. In this paper, we propose a formalization of norm conflicts based on the normative states of interrelated norms. Then via operationalizing the normative structure based on Colored Petri Nets, we propose a method for detecting such conflicts.

Keywords: Regulated organizations, Normative systems, Norm conflicts, Agent organizations

1 Introduction

A common problem for organizations is the increasing amount and complexity of norms that they have to consider in the design of their business processes. For example, when dairy products are exported, besides the internal process control of the dairy exporter, many other sources of norms are imposed by different institutions [5]. For instance, customs regulates the activities concerning export declaration, and transportation. Health agency regulates the activity of health certification. Agriculture agency puts information requirements on export declaration. Tax agency regulates the activities of Value-Added Tax settlement and invoicing. Given the diversity of regulation sources and possibly conflicting interests, it is likely that the norms imposed by these institutions are not consistent. In such cases, it is impossible to reach an agreement on whether the organizations comply with the regulations, which may cause misunderstanding and decrease the effectiveness of laws and regulations. To this end, mechanisms are needed to detect the norm conflicts.

Such a problem has been extensively investigated by researchers in the domain of normative systems. An early work is presented by [13], in which the concept of normative conflict is formally analyzed and two approaches of reasoning with normative conflicts are discussed. [16, 15] applied first-order unification to discover overlapping substitutions to the variables of laws/norms in which legal/norm conflicts may occur. Targeting distributed management of norms, [3] proposed a normative model based on the propagation of normative positions as consequences of agents' actions, and realized conflict detection by providing a mapping of the normative model into Colored Petri Nets. Focused on normative conflicts in electronic contracts, [4] presented a set of primitive conflict patterns and proposed the representation of e-contracts in default logic to facilitate conflict detection. [10, 9] proposed a computational model for detecting norm conflicts given traces of agent actions by means of Answer Set Programming. Focused on identifying conflicts between obligations in dynamic settings, [14] introduced a new semantics for the obligations to identify the necessary and sufficient conditions to detect conflicting obligations. Though these approaches provide useful formalisms and detection techniques, there are two issues that have not been discussed. One is the analysis of how the interrelations between norms might influence the existence of norm conflicts. The other one is how compliance status of norms is linked to the existence of norm conflicts.

Targeting the two issues, this paper investigates the concept of norm conflicts in the setting of interrelated norms. To formalize the specification of norms, we adopt the normative language Norm Nets (NNs) [8] which provide formalisms for representing the interrelations between norms. Based on NNs, we present an analysis of norm conflicts in terms of the compliance status of norms and show how interrelations between norms may influence the formation of norm conflicts. Moreover, we distinguish between two types of norm conflicts, i.e., weak conflicts and strong conflicts. To detect the conflicts, a computational model is developed by using Colored Petri Nets [6].

The rest of the paper is organized as follows. Section 2 introduces the formalisms that are used to model norms. Section 3.2 gives the definition of norm conflicts and presents the mechanism of detecting such conflicts. Section 4 provides a case study. Finally, section 5 concludes this paper and identifies the directions for future work.

2 Normative Structure

In this paper, we consider an institution as a set of norms used to regulate the behavior of participating agents in organizations [11], which is formalized by Norm Nets (NNs) [8].

2.1 Conceptual Model

(i) Preliminaries *Events* are defined to represent the actions available to the roles in organizations.

Definition 1 (Event). *Let R be a finite set of roles and $\mathcal{A}$ be a finite set of actions. The set of events $E \subseteq R \times \mathcal{A}$ where an element from E is denoted as $\varepsilon = (r, \alpha), r \in R, \alpha \in \mathcal{A}$.*

An event $\varepsilon = (r, \alpha)$ describes an action α available to a role r. For example, we can express an institutional observation "a student enters the library" by defining an event (*Student*, *enter_library*). Using the notion of events, a propositional language L_E is defined over the set of events.

Definition 2 (Event language). *Given an event $e \in E$, let the event language L_E be the set of expressions generated by the following grammar:*

$$\varphi ::= e|(\varphi \wedge \varphi)|(\varphi \vee \varphi)|(\varphi < \varphi)|\lambda$$

$\varphi_1 \wedge \varphi_2$ indicates both φ_1 and φ_2 occur (conjunction), $\varphi_1 \vee \varphi_2$ indicates either φ_1 or φ_2 occurs (disjunction), $\varphi_1 < \varphi_2$ indicates φ_1 occurs before φ_2 (sequence), and λ represents a null event. E_φ is used to indicate all the events contained in φ. The event expressions can be evaluated to *true* or *false* based on the occurrence of prescribed events and their relations.

For example, given $\varphi_1 = $ (*Student*, *enter_library*) and $\varphi_2 = $ (*Librarian*, *check_identity*), $\varphi_1 \wedge \varphi_2$ means that the student enters the library and the librarian checks the identity; $\varphi_1 \vee \varphi_2$ means that either the student enters the library or the librarian checks the identity; $\varphi_1 < \varphi_2$ means that the student enters the library and then the librarian checks the identity.

(ii) Norms *Norms* are defined to prescribe how agents ideally should (not) behave in terms of the roles they enact. In NNs, two types of norms are defined, i.e., obligations and prohibitions, as formalized in Definition 3.

Definition 3 (Norm). *A norm $n = (D, \rho, \delta, \sigma)$ where (1) $D \in \{\text{O}, \text{F}\}$ indicates the deontic type of the norm, i.e., Obliged, Forbidden, (2) $\rho \in E$, describing a non-empty target to which the deontic modality is assigned, (3) $\delta \in L_E$, describing the deadline of the norm, and (4) $\sigma \in L_E$, describing the precondition of the norm.*

The target is indicated by a role-action pair in which the role specifies to whom the norm applies and the action specifies the behavior that is constrained by the norm. Both the precondition and the deadline are event formulas. The precondition determines when the norm is activated and enforced, and the deadline determines when an obligation has to be ensured or a prohibition ceases.

For example, we can model a regulation that "If a student borrows a book from the library, the student should return the book within 1 month" by defining a norm $n = (O, (Student, return\_book), (Timeline, pass\_1month), (Student, borrow\_book))$. In this norm, we have defined two roles *Student* and *Timeline*, in which *Timeline* is a reserved role used to indicate the elapsing of time.

(iii) State Transitions of Norms A norm is *instantiated* when it is created. As soon as the precondition holds, the norm is *activated*. An obligation is considered *satisfied* when both its precondition and target are true while its deadline is false, and considered *violated* when both its precondition and deadline are true while the target is false. A prohibition is considered *satisfied* when both its precondition and deadline are true while its target is false, and considered *violated* when both its precondition and target are true while its deadline is false.

(iv) Norm Nets To capture the interrelations between norms, the concept of *Norm Net* is introduced.

Definition 4 (Norm net). *A norm net NN is defined by the following BNF:*

$$NN ::= n|\mathrm{AND}(NN,\ NN)|\mathrm{OR}(NN,\ NN)|\mathrm{OE}(NN,\ NN)$$

where n is a norm; S_{NN} is used to denote the set of component norm nets contained in NN and E_{NN} is used to denote the set of events contained in NN.

A norm net can be a single norm or a nested structure composed of norms with three different relations. **AND** indicates that both component norm nets should be satisfied and the violation of either component will result in a violation to the combination. **OR** indicates a choice between the two component norm nets and only when both are violated the combination is considered as violated. **OE** indicates that the two component norm nets are conditional and exclusive, i.e., (1) only when the first component is violated can the second component be activated, (2) the violation of the first component can be repaired by the second component being satisfied. Based on the state transitions of single norms, the state transitions of NNs can be derived according to the interrelations between its component norms.

For example, consider the following normative constraint "students should return the book within 1 month after they borrow the book, otherwise they have to pay a fine within 1 week." This piece of constraint indicates a reparation/sanction relation between two norms, which can be represented by a norm net $OE(n_1, n_2)$ where $n_1 = (O, (Student,\ return\_book),\ (Timeline,\ pass\_1month),\ (Student,\ borrow\_book))$ and $n_2 = (O, (Student,\ pay\_fine),\ (Timeline,\ pass\_1week),\ \lambda)$.

2.2 Operational semantics

The operational semantics of NNs are obtained by a mapping to Colored Petri Nets (CPNs) following the approach presented by [8].

(i) Colored Petri Nets A CPN [6] is a directed graph consisting of two types of nodes, called *places* and *transitions*, where *arcs* are either from a place to a transition or from a transition to a place. Tokens in CPNs may have different colors or data types and carry data attributes that characterize the entities the tokens represent.

A place serves as a placeholder for the entities in the system being modeled. Each place is associated with a type or a color set that determines the kind of tokens the place may contain. Besides a type, each place has a *marking* (denoted as M) to indicate its state, which is defined as a multiset of values over the type of the place. A *multiset* is similar to an ordinary set except that the same element can occur multiple times. A *token* is an element of such a marking, i.e., it has a value and resides in a place. If a marking consists of tokens with different values, we separate them with two pluses ($++$). The subtraction of tokens with different values is expressed as two minuses ($--$).

An arc has an *inscription* which may contain one or more free variables. Transitions represent the events that can occur in the system being modeled. A transition has a set of variables, i.e., the ones occurring on all the arcs connecting to it. Each of these variables can be assigned a value from the set represented of its type. A transition along with an assignment of each of its variables is referred to as a *binding* (denoted as b). Given a CPN with a marking and a binding, the binding is considered to be enabled if all input places contain at least the tokens specified by the evaluation of the expression on the corresponding input arcs in the binding. A transition is enabled in a marking if there exists at least one binding which is enabled in the marking. If a binding or a transition is enabled, it can occur or be fired. This results in consuming all the tokens from input places corresponding to the evaluations of the expressions on input arcs and producing new tokens on output places corresponding to the evaluations of the expressions on output arcs. The marking of a model before we start simulation is called the *initial marking*.

Based on the description above, the formalization of a CPN is shown in Definition 5.

Definition 5 (CPN). *A CPN is a tuple (P, T, A, Σ, V, C, E, I) where (1) P is a finite set of places, (2) T is a finite set of transitions such that $P \cap T = \emptyset$, (3) $A \subseteq P \times T \cup T \times P$ is a set of directed arcs, (4) Σ is a finite set of non-empty color sets (data types), (5) V is a finite set of typed variables such that $Type[v] \in \Sigma$ for all variables $v \in V$, (6) $C : P \to \Sigma$ is a color set function that assigns a color set to each place, (7) $E : A \to EXPR_V$ is an arc expression function that assigns an arc expression to each arc $a \in A$ such that $Type[E(a)] = C(p)_{MS}$, where p is the place connected to the arc a and MS indicates $C(p)_{MS}$ is a multiset, (8) $I : P \to EXPR_\emptyset$ is an initialization function that assigns a closed expression to each place p such that $Type[I(p)] = C(p)_{MS}$.*

(ii) Mapping From NNs to CPNs In organizations, roles are enacted by agents, and the agents' behavior is constrained by the norms regulating the roles they enact. In this paper, we assume a set of agents Ag participating in a regulated organization and an explicit enactment relation REA between the agents and the roles specified in the norms. Based on the definition of NNs and their state transitions, correspondences between a norm net and a Colored Petri net can be generalized as follows.

- $R \to \Sigma$: each role corresponds to a color set which can be assigned to the places,
- $\mathcal{A} \to T$: actions are represented by the transitions,
- $E \to P \times T$: events are indicated by the connections from places to transitions,
- $(REA \subseteq Ag \times R) \to I$: role-enacting agents are indicated by the initial distribution of tokens in the places (i.e., initial marking),
- *Satisfied* $\subseteq P$, *Violated* $\subseteq P$, *Satisfied* $\cap$ *Violated* $= \emptyset$: the satisfied and violated states of norms are indicated by two disjoint subsets of the places.

Agents are represented by the tokens which can only reside in the places with the matching colors according to the roles the agents are enacting.

Based on the correspondences between the elements in NNs and that in CPNs, we follow the approach presented by [8] and the CPN patterns in [12] to construct the CPN model of NNs. The resulting CPN model of a norm net is given as follows.

Definition 6 (CPN model of NNs). *The CPN model of a norm net NN with an enactment relation REA is denoted as $\Theta(NN, REA) = (\mathcal{N}$, Satisfied, Violated, $p_s, p_v)$ where (1) $\mathcal{N}$ is a CPN according to Definition 5, (2) Satisfied $\in P_{\mathcal{N}}$ is a subset of the places of $\mathcal{N}$, indicating the satisfied states of all the component norm nets in NN, (3) Violated $\in P_{\mathcal{N}}$ is a subset of the places of $\mathcal{N}$, indicating the violated states of all the component norm nets in NN, (4) $p_s \in$ Satisfied is a place of $\mathcal{N}$ such that $\nexists t \in T_{\mathcal{N}} : (p_s, t) \in A_{\mathcal{N}}$, indicating the overall satisfied state of NN, (5) $p_v \in$ Violated is a place of $\mathcal{N}$ such that $\nexists t \in T_{\mathcal{N}} : (p_v, t) \in A_{\mathcal{N}}$, indicating the overall violated state of NN.*

3 Norm Conflicts

Based on the formalism of NNs, in this section, we propose a definition of norm conflicts in terms of the compliance status of norms. Taking into account the interrelations between norms, the definition gives a comprehensive representation of norm conflicts. Furthermore, a computational model is developed to detect the conflicts.

3.1 Definition

A conflict occurs between an obligation and a prohibition when they constrain on the same behavior and have an overlapped activation period (cf. [16]). That is, if some behavior of the same role is obliged and forbidden at the same time, a conflict arises. From this definition, we differentiate between two types of norm conflicts. First, *weak conflicts*: the activation period of the prohibition does not cover the whole activation period of the obligation. In this sense, a weak conflict can be avoided when the event constrained by the two norms, occurs during the time period when the obligation is activated while the prohibition is not. In this way, both norms can be satisfied. Second, *strong conflicts*: the activation period of the prohibition covers the whole activation period of the obligation. That is, whenever the event constrained by the two norms occurs, or whether or not it occurs, one of the norms will be violated. In essence, a conflict occurs between an obligation and a prohibition when the two norms cannot be satisfied at the same time, i.e., the compliance status of the two norms is evaluated to be contradictory with respect to the occurrence of an event.

Furthermore, to determine whether a norm conflict exists, there is another criterion that has to be considered, i.e., the compliance relation between norms. In section 2, we have introduced three compliance relations between norms, i.e., AND, OR and OE. Therefore, with an event occurring, if two (or more) norm nets with contradictory compliance evaluations are combined, the conflicting status of the combined norm net depends on the compliance relations of the component norm nets. If the compliance relation is AND, a conflict occurs since the combined norm net cannot reach an agreement on the compliance of the event. If the compliance relation is OR, there is no conflict since the combined norm net only picks up the positive evaluation result, i.e., satisfied. As for the compliance relation of OE, there is never a conflict since the activation period of the two component norm nets will never overlap, i.e., only when the origin is violated can the reparation be activated. Therefore, a norm conflict between two norm nets may occur only when the two norm nets are connected by an AND compliance relation.

Based on the description above, we give the definition of a *norm conflict* as follows.

Definition 7 (Norm conflict). *Given the occurrence of an event e, a norm conflict arises in a norm net NN iff $\exists$ NN_x, NN_y $\in$ $S_{NN}(NN)$ such that (1) NN_x and NN_y have an AND compliance relation, and (2) NN_x is evaluated to be satisfied and NN_y is evaluated to be violated.*

Given the occurrence of an event, a conflict occurs in a norm net when there are two AND-related component norm nets in the norm net whose normative states are respectively evaluated to be satisfied and violated. While for the conflicting component norm nets themselves, they may have a nested structure of norms connected by other compliance relations such as OR and OE.

It can be seen that our definition of norm conflicts is from the perspective of norm compliance, which is different from other definitions provided in the literature (e.g., [16], [10]). The advantage of our definition is that it can be easily extended to other types of norms or normative structures since it captures the root cause of norm conflicts. For example, it is possible that a role-enacting agent is regulated by both an obligation to sit and another obligation to stand whose activation period have an overlap. In this case, a conflict occurs since the two actions “sit” and “stand” are physically exclusive to each other, which can be reflected from the compliance evaluation results of the two norms. While, if defining a norm conflict at the level of norm specification between an obligation and prohibition, such conflicts may not be covered. The differentiation between weak and strong conflicts is based on whether there are possible event sequences that can avoid introducing conflicting normative states (i.e., satisfied and violated), which will be detailed in Section 3.3. Moreover, we take into account the impact of the compliance relations between norms.

3.2 Detection

Given the definition above, we now illustrate how to make use of the CPN models of NNs to computationally detect the norm conflicts. To do this, there are three steps. The *first* step is to construct the CPN model of the norm net NN, following the procedure presented in [8]. The *second* step is to obtain the new marking of the CPN model with respect to the occurrence of the enabled transitions given the event e. Comparing the new marking with the previous marking, we can derive the changes of the normative state of all the component norm nets in NN by looking at the satisfied places and the violated places. The *third* step, including two sub-steps, is to determine whether there is any norm conflict in NN with respect to the occurrence of the event e. The first sub-step checks whether NN is evaluated to be violated. If so, the second sub-step is to further check whether there are any two component norm nets in NN that are respectively evaluated to be satisfied and violated, by looking at the token distribution in the places representing the satisfied and violated states. Algorithm 1 gives the procedure of detecting norm conflicts in a norm net with respect to the occurrence of an event.

The problem of detecting whether a sequence of events will cause any conflicts in a norm net can be transformed into the problem of pattern matching of CPN markings/states, the complexity of which is shown to be $O(L \cdot W^2)$ where L is the size of the event sequence and W is the size of the CPN model (i.e., the number of nodes in the CPN model).

3.3 Weak and Strong Conflicts

We have shown the mechanism of detecting norm conflicts using the CPN models of NNs. Now we continue with the question of whether a norm conflict found in a norm net is a weak or strong conflict. To this end, we assume the set $\aleph$ of all the possible instances of a norm net (i.e., all the possible states of the real system) and give the following definition.

Definition 8 (Weak conflict). *A weak conflict is detected in a norm net NN with respect to an event e iff*

1. *there exists an instance of NN from the set of all possible instances $\aleph$ such that a norm conflict exists in the instance, and*
2. *there exists an instance of NN from the set of all possible instances $\aleph$ such that no norm conflict exists in the instance.*

The first condition indicates that there exists a norm net instance of NN in which a conflict is found with respect to the occurrence of the event e. The second condition indicates that there exists a norm net instance of NN in which no conflict is found with respect to the occurrence of the event e.

In a similar way, a strong conflict is defined as follows.

Definition 9 (Strong conflict). *A strong conflict is detected in a norm net NN with respect to an event e iff for every instance of NN from the set of all possible instances $\aleph$, there is always a norm conflict existing in the instance according to Definition 7.*

The condition of a strong conflict indicates that for every possible instance of the norm net NN there is always a conflict found with respect to the occurrence of the event e. Given the definition of a weak conflict and strong conflict, we define a consistent norm net as follows.

Algorithm 1 Conflict Detection

Require: (NN, REA, e) ▷ A norm net with en enactment relation and an event
Ensure: CFS ▷ Conflicting status

▷ Obtain the enabled bindings of the CPN model $\mathcal{N}$ given the occurrence of event e and the current marking M
function EnabledStep($e, M, \mathcal{N}$)
  $Y \leftarrow \emptyset$
  $(r, \alpha) \leftarrow e$
  for all $(p, t) \in A_{\mathcal{N}}$ **do**
    if $C(p) = r$ and $t = \alpha$ and $E(p, t)\langle b\rangle \leq M(p)$ **then**
      $Y \leftarrow Y \cup (t, b)$
    end if
  end for
return Y
end function

▷ Obtain the new marking of the CPN model $\mathcal{N}$ given the occurrence of event e and the current marking M
function UpdateState($Y, M, \mathcal{N}$)
  for all $p \in P_{\mathcal{N}}$ **do**
    $M'(p) \leftarrow M(p) -- \left(\sum^{++}_{MS}\right) \sum_{(t,b)\in Y} E(p, t)\langle b\rangle ++ \left(\sum^{++}_{MS}\right) \sum_{(t,b)\in Y} E(t, p)\langle b\rangle$
  end for
  $Y \leftarrow \text{EnabledStep}((*, null), \mathcal{N}, M')$
  if $Y \neq \emptyset$ **then**
    $M' \leftarrow \text{UpdateState}(Y, M', \mathcal{N})$
  end if
return M'
end function

▷ (**Step 1**) Obtain the CPN model of the norm net NN with the role enactment REA
$(\mathcal{N}, \textit{Satisfied}, \textit{Violated}, p_s, p_v) \leftarrow \Theta(NN, REA)$

▷ (**Step 2**) Obtain the new normative state of NN given the occurrence of the event e
$M \leftarrow I_{\mathcal{N}}$
$Y \leftarrow \text{EnabledStep}(e, M, \mathcal{N})$
$M' \leftarrow \text{UpdateState}(Y, M, \mathcal{N})$

▷ (**Step 3**) Check the normative state changes of all the component norm nets in NN
  ▷ (**Step 3.1**) Check whether the normative state of NN is evaluated to be violated
$CFS \leftarrow false$
if $M'(p_v) - -M(p_v) > 0$ **then**
  for all $(p, p') \in \textit{Satisfied} \times \textit{Violated}$ **do**
    ▷ (**Step 3.2**) Check whether there are two component norm nets in NN such that one is evaluated to be satisfied and the other is evaluated to be violated
    if $(M'(p) -- M(p)) > 0$ **and** $(M'(p') -- M(p')) > 0$ **then**
      $CFS \leftarrow true$
    end if
  end for
end if

Definition 10 (Consistent norm net). *A norm net NN is consistent iff $\forall e \in E_{NN}$, neither a weak conflict nor a strong conflict is detected in NN.*

A norm net is consistent if and only if the occurrence of any event specified in the norm net does not lead to a norm conflict (weak and strong).

The complexity of determining whether the occurrence of an event causes a weak conflict or a strong conflict in a norm net is $O(W^2 \cdot V)$ where W is the size of the CPN model of the norm net and V is the number of nodes in the state space of the CPN model. That is, in the worst case, we have to search over the complete state space of the CPN model.

4 Case Study

4.1 Case Description

The World Customs Organization has defined a framework called the Authorized Economic Operator (AEO) program [2] in order to address the tensions created by the simultaneous growth in international trade and requirements for increased security. The European Communities' implementation of AEO permits various customs administrations to grant AEO certificates to qualified companies under which they enjoy special privileges. Taking the scenario of importing food from a country outside the EU to the Netherlands, a number of governmental authorities and companies are involved, which are governed by different sets of regulations concerning different aspects of the food importation process. For example, the EU has a set of general regulations, one of which specifies that the food authority is *obliged* to carry out a food quality inspection. With the introduction of the AEO programme, the Dutch government introduced new regulations for the specific domain of AEO-certified goods in order to improve trading efficiency. For example, one regulation specifies that a food authority is *forbidden* to carry out a food quality inspection, if the customs has already done so. Additionally, companies such as *container terminals* play an important role and bring their own regulations, e.g., a regulation at one container terminal is that carriers are *obliged* to transport their goods thence within two days of unloading. With different values and interests, the regulations from these institutions are likely to be inconsistent.

4.2 Modeling Norms

In this case study, we consider three institutions $\mathcal{I}_1$, $\mathcal{I}_2$ and $\mathcal{I}_3$ respectively corresponding to the regulation of EU, Dutch government and a Container terminal, captured by three norm nets NN_1, NN_2, and NN_3 described as follows.

- $NN_1 = \text{AND}(\text{AND}(n_{11}, n_{12}), n_{13})$ where $n_{11} =$ (O, (*Food_authority*, *inspect_quality*), (*Carrier*, *transport_goods*), (*Carrier*, *arrive*)); $n_{12} =$ (F, (*Carrier*, *unload_food*), (*Food_authority*, *inspect_quality*), λ); $n_{13} =$ (F, (*Carrier*, *choose_inspectLocation*), λ, λ))

- $NN_2 = \text{AND}(n_{21}, n_{22})$ where
  n_{21} = (F, (*Food_authority*, *inspect_quality*), λ, (*Customs*, *inspect_quality*));
  n_{22} = (F, (*Carrier*, *unload_food*), (*Food_authority*, *inspect_quality*), λ)

- $NN_3 = \text{OE}(n_{31}, n_{32})$ where
  n_{31} = (O, (*Carrier*, *transport_goods*), (*Timeline*, *pass_2days*), (*Carrier*, *arrive*)); n_{32} = (O, (*Carrier*, *pay_fine*), (*Timeline*, *pass_1month*), λ)

Four roles are defined in the three norm nets, i.e., *Carrier*, *Food_authority*, *Customs* and *Timeline*. In particular, *Timeline* is a reserved field for representing the pass of time. At the moment, we assume an equal status of regulation between different institutions, i.e., the norm nets representing the three institutions are combined with an AND relation, represented as $\text{AND}(\text{AND}(NN_1, NN_2), NN_3)$. We will explore more advanced relations between institutions such as priority relation in future work.

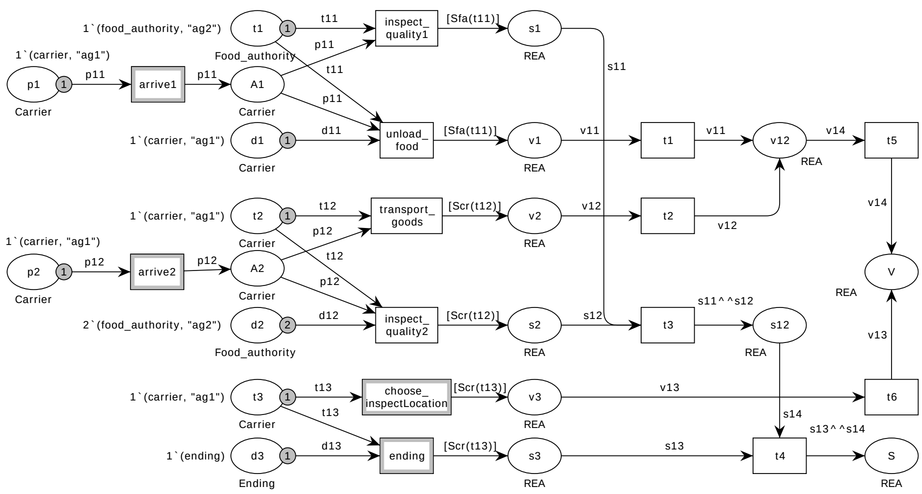

Fig. 1. The CPN model of NN_1.

In this case study, we assume three agents ag_1, ag_2, ag_3 respectively enacting the roles *Carrier*, *Food_authority*, *Customs*. To operationalize the regulation of the three institutions, we build for each norm net a CPN model following the approach presented by [8]. As an example, we show the CPN model of NN_1 in Figure 1. Places are drawn as ellipses and transitions are drawn as rectangles. Enabled transitions are indicated by bold outlines. The color assigned to each place is indicated by the label below the ellipses. The action each transition represents is indicated by the label inside the rectangles. Role-enacting agents are represented by the dots with a number inside. For example, there is a token valued 1‘(*carrier*, “*ag1*”) in place *p1*, representing an agent named *ag1* enacting the role *carrier*. The satisfied and violated states of all the component norm nets are indicated by the places whose labels start with s and v. Specifically, a color set *REA* is assigned to all these satisfied and violated places/states in the CPN model, which is defined as a union of the set of all the roles specified in the corresponding norm net, indicating that any role-enacting agents may satisfy or violate the norms. The color set *Ending* together with the transition *ending* is defined specifically to signal the ending of an event sequence such that norms whose deadline is null can be evaluated accordingly, e.g., the prohibition n_{13} .

4.3 Detecting Conflicts

Figure 2 shows a part of institutional evolutions with respect to three sequences of events in this case study. In general, when an event occurs (shown above/below the arrows), the system will identify which norms in the relevant institutions are triggered/activated. Each circle represents a normative state of the three institutions. With more than one norm from different institutions being triggered simultaneously, conflicts might occur. For example, three norms from the three institutions are triggered simultaneously at M_3, between which two conflicts occur. In this case study, there are in total three pairs of conflicts (indicated by a line with a cross).

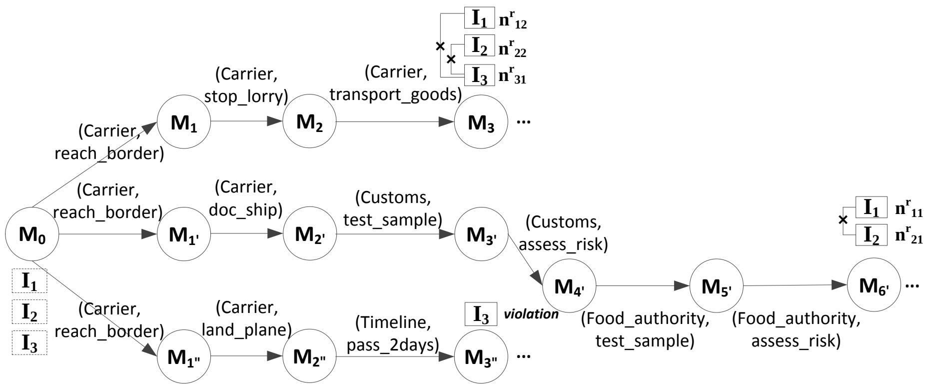

Fig. 2. Institution evolution and norm conflicts.

As an example, we show how the normative states of the three institutions change along with the first event sequence in terms of the markings of the corresponding CPN models implemented in CPN tools [7], as shown in Figure 3. There are three markings, denoted as nodes 1, 2 and 3, each of which is indicated by the number of tokens each place of the CPN models contains, as listed in the box under each node. It can be seen that initially at the marking represented by node 1, all the overall satisfied and violated states (places labeled with S and V) of the three institutions are empty, highlighted by the red rectangles. From node 1, an event (*Carrier*, *arrive*) occurs. As a result, another marking represented by node 2 is generated, in which the states of the three pairs of satisfied and violated places remain the same as that at node 1. From node 2, an event (*Carrier*, *transport_goods*) occurs. As a result, we obtain a new marking represented by node 3, in which the overall violated places of the first and second institutions $Institution1'V$ and $Institution2'V$ both get a token (*carrier*, "*ag1*") while the overall satisfied places $Institution1'S$ and $Institution2'S$ remain empty, indicating that the agent $ag1$ enacting the role $Carrier$ produces a violation in both $\mathcal{I}_1$ and $\mathcal{I}_2$. However, the overall violated place of the third institution $Institution3'V$ remains empty but the overall satisfied place $Institution3'S$ gets a token (*carrier*, "*ag1*"), indicating that $\mathcal{I}_3$ is satisfied with respect to the agent behavior. These imply that the three institutions give contradictory compliance evaluation results with respect to the same event. Therefore, we can derive two norm conflicts, respectively between $\mathcal{I}_1$ and $\mathcal{I}_3$, and between $\mathcal{I}_2$ and $\mathcal{I}_3$.

5 Conclusions

In this paper, we provide an analysis of norm conflicts in the setting of interrelated norms. From the perspective of compliance, we show how norm conflicts can be modeled and analyzed. Moreover, we consider the influence of compliance relations between norms with respect to norm conflicts, which provides an integrated view on the analysis of sets of norms. To operationalize conflict detection, we make use of the CPN models of NNs and show the application of the state space analyzing tools from CPNs in detecting (potential) conflicts. Our definition of norm conflicts is not bound to a specific computational mechanism but can also be combined with other detection techniques, e.g., the model based on Answer Set Programming proposed by [10]. However, the choice of CPNs is supported by its capability of modeling concurrent systems, graphical expressions, tool support and advanced state space analyzing techniques. Moreover, variants of CPNs such as hierarchical CPNs, timed CPNs and stochastic CPNs, provide extended support for the analysis of norm conflicts when aspects such as levels, time and probability are considered.

There are several directions for future work. Firstly, we will further investigate the interrelations between norms and their impact on norm conflicts. For example, norms may have different priority in the regulation of organizational behavior, and the fulfillment of a norm may cancel the enforcement of some other norms. In practice, norms are not always specified at a single level of abstraction, which necessitates research on norm conflicts between norms at multiple abstraction levels. Given the results of norm conflicts, an important question is how to resolve such conflicts. To this end, we are going to investigate approaches that can learn from agents' past behavior and find optimal solutions for norm revision in terms of, e.g., the number of norms to be changed, the number of roles involved, the overall social risk and welfare, etc.

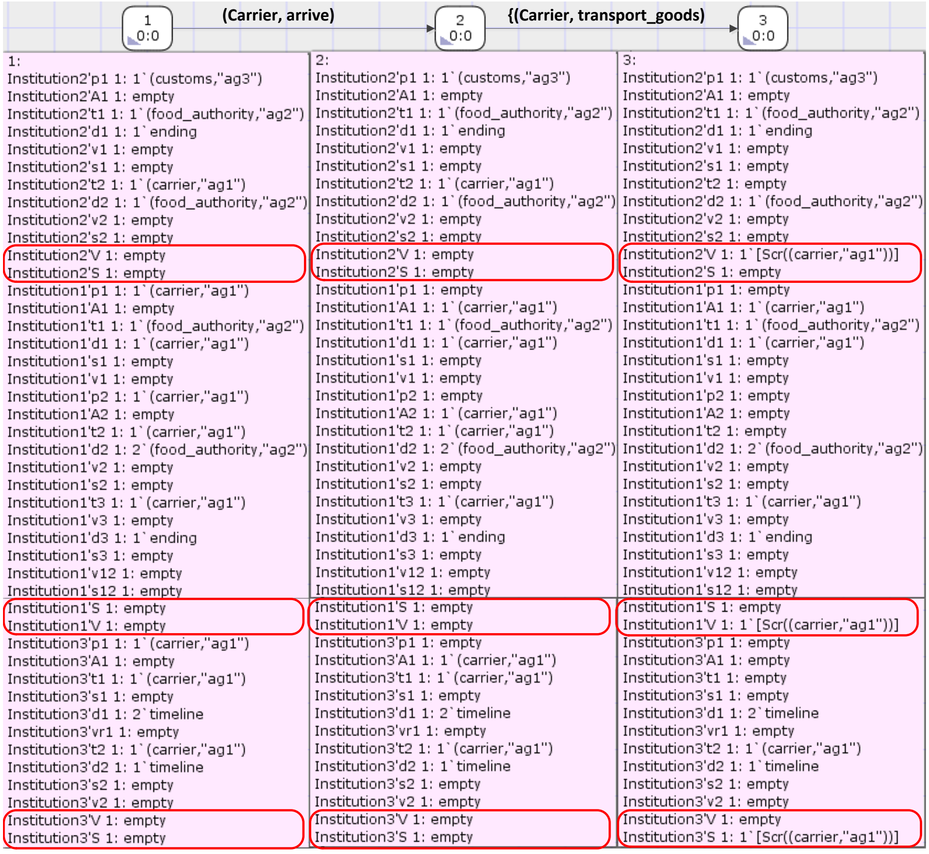

Fig. 3. Normative state transition of the three institutions.

References

1. Allan Cheng, Søren Christensen, and Kjeld Høyer Mortensen. Model checking coloured petri nets exploiting strongly connected components. In *International Workshop on Discrete Event Systems*, 1996.
2. European Commission. Authorised economic operator. `http://ec.europa.eu/taxation_customs/customs/policy_issues/customs_security/aeo/`, 2013.
3. D. Gaertner, A. Garcia-Camino, J. A. Rodriguez-Aguilar, and W. Vasconcelos. Distributed norm management in regulated multi-agent systems. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, pages 624–631, 2007.
4. Georgios K. Giannikis and Aspassia Daskalopulu. Normative conflicts in electronic contracts. *Electronic Commerce Research and Applications*, 10(2):247–267, 2011.
5. S. Henningsson, N. Bjørn-Andersen, A. Schmidt, B. Flügge, and Z. Henriksen. Food living lab – complexity of export trade. In Y. H. Tan, N. Bjorn-Andersen, S. Klein,

and B. Rukanova, editors, *Accelerating Global Supply Chains with IT-Innovation.* Berlin: Springer-Verlag, 2011.

6. Kurt Jensen. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Uses.* Springer, 1997.
7. Kurt Jensen, Lars M. Kristensen, and Lisa Wells. Coloured petri nets and cpn tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer*, 9(3-4):213–254, 2007.
8. Jie Jiang, Huib Alderwereld, Virginia Dignum, and Yao Hua Tan. Compliance checking of organizational interactions. *ACM Transactions on Management Information Systems*, 5(4), 2015.
9. T. Li, J. Jiang, H. Aldewereld, M. De Vos, V. Dignum, and J. Padget. Contextualized institutions in virtual organizations. In *Coordination, Organisations, Institutions and Norms in Agent Systems VIIII*, 2013.
10. Tingting Li, Tina Balke, Marina De Vos, Ken Satoh, and Julian Padget. Detecting conflicts in legal systems. In *New Frontiers in Artificial Intelligence*, pages 174–189, 2013.
11. Elinor Ostrom. *Understanding institutional diversity.* Princeton University Press, 2005.
12. Nick Russell, Arthur H. M. ter Hofstede, Wil M. P. van der Aalst, and Nataliya Mulyar. Workflow control-flow patterns: A revised view. Technical Report BPM-06-22, BPM Center, 2006.
13. G. Sartor. Normative conflict in legal reasoning. *Artificial Intelligence and Law*, 1:209–235, 1992.
14. Silvano Colombo Tosatto, Guido Governatori, and Pierre Kelsen. Detecting deontic conflicts in dynamic settings. In *Deontic Logic and Normative Systems*, Lecture Notes in Computer Science, pages 65–80. Springer, 2014.
15. W. Vasconcelos, M. Kollingbaum, and T. Norman. Normative conflict resolution in multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 19(2):124–152, 2009.
16. W. Vasconcelos, M. J. Kollingbaum, and T. J. Norman. Resolving conflict and inconsistency in norm-regulated virtual organizations. In *Proceedings of 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 644–651, 2007.