# How to Model Normative Behavior in Petri Nets


Jean-François Raskin
Computer Science Institute
Fac. Univ. Notre-Dame de la Paix
Rue du Grandgagnage 21
B5000 Namur
Belgium
jfr@info.fundp.ac.be

Yao-Hua Tan[1]
Leendert W.N. van der Torre[1,2]
[1]EURIDIS
[2]Dept. of CS and Tinbergen Inst.
Erasmus University Rotterdam
P.O. Box 1738, 3000 DR Rotterdam,
The Netherlands
{ytan,ltorre}@euridis.fbk.eur.nl



**Abstract**

In this paper, we show how to extend the Petri net formalism to represent different types of behavior, in particular normative behavior. This extension is motivated by the use of Petri nets to model bureaucratic procedures, which contain normative aspects like obligations and permissions. We propose to extend Petri nets with a preference relation, a well-known mechanism from deontic logic to discriminate between ideal and varying sub-ideal states.


## 1 Introduction

Petri nets [Pet81] are a popular formalism for the modeling and analysis of discrete dynamic (distributed) systems, because they combine the advantages of a graphical representation with the expressive power of parallelism and synchronization. One of the application domains is the modeling of procedures and processes within and between organizations. For example, van der Aalst [vdA92] developed the ITCPN model to model logistic processes in organizations, and Lee [Lee91, Lee92] developed the CASE/EDI tool to model bureaucratic procedures. The latter tool can be used to dynamically simulate Petri nets (scenario analysis) and check the procedures represented in a Petri net for consistency and dead-locks. It has been applied to model inter-organizational procedures in international trade, like contract negotiation, the exchange of bill-of-lading for letter-of-credit and custom clearance [BLWW95]. This paper is not devoted to the defense of the use of Petri nets for modeling processes or procedures (this has been defended in the papers mentioned above). Also, we do not claim that our deontic Petri nets are more appropriate to model normative behavior than deontic logics. The only claim we make is that if one uses Petri nets to model procedures, and one also wants to model the deontic aspects of these procedures in the nets, then the paper provides a technique using which this can be achieved.

In this article, we argue that in Petri nets deontic aspects of procedures should be represented by a preference ordering on executions of the nets. We borrow the idea of representing deontic aspects by preferences from deontic logic, a modal logic where the modal formula $O(p)$ means that $p$ is obliged. Deontic logics with preferential semantics [Han71, PS94, TvdT95, TvdT96] can represent so-called Contrary-To-Duty (CTD) obligations, because the preferential semantics can distinguish between ideal and varying sub-ideal behavior. In this way they solve the so-called deontic paradoxes caused by CTD obligations [Chi63, For84]. It is known from the area of knowledge representation (see e.g. [JS92]) that deontic logic is a useful knowledge representation language when the modeler wants to represent sub-ideal states and CTD obligations. It is important that violations of obligations, i.e. sub-ideal states, are represented explicitly in the modeling of procedures, because in most procedures it is described explicitly what is considered as ill-behavior, and how this will be punished (the corresponding sanction). Since this violation behavior is described explicitly, it should also be represented explicitly in the Petri net. Representing sub-ideal behavior does not make sense if there is no way to represent ideal behavior in the Petri net, just as the notion of slave does not have a meaning without there being a master. Hence, both ideal and sub-ideal behaviors must be represented and distinguished from each other in Petri nets for modeling procedures.

The layout of this article is as follows. Section 2 introduces the Petri net formalism and a running example from the well-known library domain. Section 3 extends the Petri net formalism with preferences in such a way that normative behavior can be modeled. Section 4 links the introduced preferences to deontic notions like obligations and permissions. Section 5 draws some conclusions and mentions several future research directions.

## 2 The Petri net formalism

A Petri net is a directed graph which consists of two disjunct sets of nodes. These nodes are called *places* (represented as circles) and *transitions* (represented as bars). Places and transitions are connected by arcs. It is not allowed to connect two places or two transitions. Arcs can have a value which indicates how many tokens are required to fire a transition. However, in the examples in this paper we only consider arcs that require exactly one token. The dynamic behavior of the modeled system is represented by tokens flowing through the net. A token is represented by a dot. Each place may contain several tokens (the so-called marking of the place). A transition is enabled if all its input places (for which there exists an arc from the place to the transition) contain the specified number of tokens, which corresponds to the value of the arc. A transition may fire whenever it is enabled. Whenever a transition is fired, it has the effect that the specified number of tokens of its input places is removed and at the same time the specified number of tokens is added to its output places. Note that we say that an enabled transition 'may' fire, because when two transitions are enabled at the same marking, we have the choice to fire one of the two. The second transition will not necessarily still be enabled in the marking which results from firing the first transition. In this way, choice is modeled in Petri nets.

We first start with the basic definitions of the Petri net formalism, see [Pet81, Gen86, vdA92] for the details. A Petri net is a directed labeled graph, in which the places are connected by transitions.

**Definition 1 (Petri net)** *A Petri net (structure) $N = (P, T, Pre, Post)$ consists of two disjunct non-empty finite sets of places $P$ and transitions $T$, and two functions $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$, where $\mathbb{N}$ stands for the set of positive integer numbers.*

The state of the graph is called a marking.

**Definition 2 (Marking)** *Let $N = (P, T, Pre, Post)$ be a Petri net. A marking $M : P \to \mathbb{N}$ is an assignment of tokens to the places of a Petri net. The marking $M$ can also be written* *as a vector $M = \langle M_1, M_2, ..., M_n \rangle$ where $n = \mid P \mid$ and each $M_i \in \mathbb{N}$, $i = 1, ..., n$. We write $\langle N, M \rangle$ for a Petri net $N$ with marking $M$.*

The marking of a graph determines which transitions are enabled.

**Definition 3 (Enabled transition)** *Let $N = (P, T, Pre, Post)$ be a Petri net with marking $M$, $M(p)$ the number of tokens contained in $p \in P$ and $t \in T$ a transition. The transition $t$ is enabled in $\langle N, M \rangle$ if and only if (iff) $\forall p \in P : M(p) \geq Pre(p, t)$.*

The dynamic behavior of a Petri net is expressed by changing markings. A marking changes when a transition fires, and a transition may fire when it is enabled.

**Definition 4 (Firing a transition)** *Let $N = (P, T, Pre, Post)$ be a Petri net with marking $M_1$, $M_1(p)$ the number of tokens contained in $p \in P$ and $t \in T$ an enabled transition. Firing the transition $t$ in $\langle N, M_1 \rangle$ results in a new marking $M_2$, written as $M_1 \rightarrow^t M_2$, given by $M_2(p_i) = M_1(p_i) - Pre(p_i, t) + Post(p_i, t)$. Hence, $Pre(p_i, t)$ denotes the number of tokens needed in $p_i$ for the firing of transition $t$ and $Post(p_i, t)$ denotes the number of tokens added to place $p_i$ when transition $t$ has fired.*

The procedural semantics of a marked Petri net is given by the set of its possible executions (sequences of transitions)[1] between two markings. We give a recursive definition of an execution.

**Definition 5 (Execution)** *Let $N = (P, T, Pre, Post)$ be a Petri net with marking $M$, $s = \langle s_1, s_2, ..., s_n \rangle$ with $s_i \in T$ a finite sequence of transitions of $T$, and $T^*$ the set of all finite sequences that can be composed from transitions of $T$. The sequence $s$ is an execution of the marked net $\langle N, M_1 \rangle$ iff it is firable in the marked net $\langle N, M_1 \rangle$, resulting in some marking $M_2$, written as $M_1 \rightarrow^s M_2$. $M_1 \rightarrow^s M_2$ iff:*

1. *either $s = \lambda$ (the empty sequence), then $M_2 = M_1$,*
2. *or $s = s't$, with $s' \in T^*$ and $t \in T$, and there is an $M_3$ such that and $M_1 \rightarrow^{s'} M_3$ and $M_3 \rightarrow^t M_2$.*

*We write $E(N, M_1, M_2) = \{s \in T^* \mid M_1 \rightarrow^s M_2\}$ for the set of possible executions of a Petri net $N$ from $M_1$ to $M_2$.*

The expressiveness of Petri nets can be enhanced by adding so-called colors to the tokens. Such Petri nets are called colored Petri nets, see e.g. [vdA92]. A colored Petri net associates with every token a color. The enabling of a transition can depend upon the colors of the tokens subsumed. In this paper, we do not use coloring for this purpose.

**Definition 6 (Colored Petri net with Marking)** *A colored Petri net (structure) $CN = (P, T, Pre, Post, C, F)$ consists of a Petri net structure $N = (P, T, Pre, Post)$, a set of colors $C$, and a transition function $F$. A colored marking $M : P \times C \rightarrow \mathbb{N}$ is an assignment of*

[1] This definition of execution is known as trace semantics. Trace semantics do not model true concurrency, in which an execution is defined as a sequence of markings (for a discussion on this issue, see [MP92]). With such a more complicated definition of execution, the ideas in this paper still apply.

*colored tokens to the places of a Petri net. The transition function $F : M \times T \rightarrow M$ relates the colors of the markings when a transition fires.*[2]

*The colored marking $M$ can also be written as a vector $M = \langle M_1, M_2, ..., M_n \rangle$ where $n = \mid P \mid$ and each $M_i = \langle M_{i,1}, M_{i,2}, \ldots, M_{i,m} \rangle$, $i = 1, ..., n$, $m = \mid C \mid$ and $M_{i,j} \in I\!N$. When we do not care about the colors of the tokens, we denote the colored marking with a standard marking such that $M_i = \sum_{j \in C} M_{i,j}$.*

The definition of an enabled transition remains the same, firing a transition changes in the sense that the transition function $F$ determines the new colors when a transition has fired. The Petri net in Figure 1 illustrates the definitions.

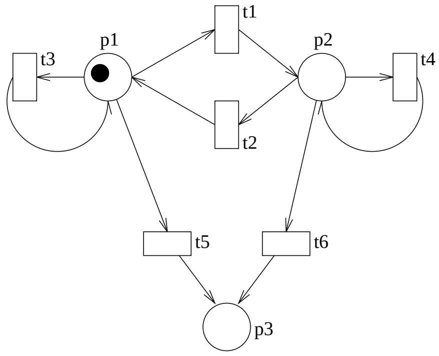


Figure 1: Borrowing books. *Places*: $p_1$: borrowed book; $p_2$: damaged book; $p_3$: returned book. *Transitions*: $t_1$: to damage the book; $t_2$: to repair the book; $t_3$, $t_4$: 1 week too late; $t_5$, $t_6$: to return the book.

The Petri net models the possible behaviors of a person who borrows books from a library. The token in place $p_1$ represents a book and moving a token from one place via a transition to another place represents behavior of the borrower. In the figure, the borrower has the choice between returning the book, keeping the book in her possession until a dead-line (of a week) has passed and damaging the book. If she decides to keep the book until the dead-line has passed, then she has again the same choices. Hence, by moving the token from place $p_1$ via $t_3$ back to $p_1$, the same situation arises. If she damages the book, then she has the choice between repairing it before returning, returning it damaged, and keeping it in her possession until another dead-line has passed. In this paper we analyze executions of this Petri net which start in marking with a single token in place $p_1$ and result in marking with a single token in place $p_3$, which represents that the book is returned. First, assume this net is a standard Petri net. In that case the structure is given by $N = \langle \{p_1, p_2, p_3\}, \{t_1, t_2, t_3, t_4, t_5, t_6\}, Pre, Post \rangle$ and a marking for this net is denoted by a tuple $\langle n_1, n_2, n_3 \rangle$ where $n_i$ indicates the number of tokens at place $p_i$. The dot in place $p_1$ represents a token at place $p_1$. Hence, the initial marking in Figure 1 is given by $\langle 1, 0, 0 \rangle$. In marking $M_1 = \langle 1, 0, 0 \rangle$, the transitions $t_1$, $t_3$ and $t_5$ are enabled. On the other hand, the transitions $t_2$, $t_4$ and $t_6$ are not enabled, because $Pre(t_2, p_2) = Pre(t_4, p_2) = Pre(t_6, p_2) = 1$ and $M_1(p_2) = 0$. Firing the transition $t_1$ in

[2] For a complete definition of the result of firing a transition, we have to specify the transition function in more detail. In particular, we have to explicitly represent the 'Pre' arcs of a transition and define the colors of the 'Post' arcs as a function of the 'Pre' arcs and the colors of the subsumed tokens. We do not further specify the transition function, because in this article we do not refer to the transition function in the remainder of the definitions.

marking $M_1$ changes the marking of the net in $\langle 0,1,0\rangle$. In the latter marking, the execution $s = \langle t_2, t_5\rangle$ can be performed, because $\langle 0,1,0\rangle \rightarrow^{t_2} \langle 1,0,0\rangle$ and $\langle 1,0,0\rangle \rightarrow^{t_5} \langle 0,0,1\rangle$.

The problem with this Petri net is that, for example, the difference between a book being on time or too late by a number of weeks is not represented in the markings of this net. Each time we apply transition $t_3$ to the initial state $\langle 1,0,0\rangle$, we end up in exactly the same marking $\langle 1,0,0\rangle$. If we want to have different markings for these time states, then we can add colors to express this difference in the markings in the following way. Let $CN = \langle P, T, Pre, Post, C, F\rangle$ be the colored Petri net such that $P$, $T$, $Pre$, $Post$ are as before, the set of colors $C$ is the set of natural numbers and the transition function $F$ is such that the color of a token is increased by one when transition $t_3$ or $t_4$ is fired. Moreover, the initial marking is given by a single token in place $p_1$ with color 0. Now the coloring of the token distinguishes exactly the different markings representing the states of any number of weeks too late. Another solution to represent this distinction in the markings could have been to introduce $n$ extra places, one for every $n$-weeks too late. So, $p_1$ represents that the book is on time and $t_{3a}$ would bring the token to another place $p_{1a}$ that represents that the book is one week too late. From this $t_{3b}$ would bring the token to another place $p_{3b}$ that represents that the book is two weeks too late etc. This solution, however, has two obvious disadvantages. First, that the net explodes. Secondly, since you do not now beforehand what the maximum number of weeks too late will be, you do not know how large to choose this $n$ beforehand.

In the introduction, we already mentioned that it is useful to model deontic aspects if the modeler wants to represent sub-ideal states and so-called contrary-to-duty (CTD) obligations. To illustrate these notions, consider two obligations that might hold for a library domain: (1) borrowed books should not be damaged and (2) if a book is damaged, then it should be repaired. The state in which the book is damaged is a sub-ideal state because a violation has occurred, and the second obligation is a CTD obligation, because it is conditional on a violation. Hence, the explicit representation of deontic aspects of this bureaucratic procedure is useful when the modeler wants to discriminate between ideal behavior and the sub-ideal behavior of damaging books, and he wants to represent the CTD obligation to repair the book. When executing this Petri net, there is the choice between performing ideal and sub-ideal behaviors. For example, if the borrower returns the book in time and not damaged, we can say that she performs ideal behavior. On the other hand, if she returns the book one week too late, then she does not perform ideal behavior. The distinction between ideal and sub-ideal cannot be represented in standard Petri nets as Figure 1. One can model the choice, but nothing in the Petri net formalism indicates that an execution is better than another. What is represented in Figure 1 is actually, one could say, only one half of the representation of sub-ideal states. Sub-ideality is partially represented in this figure to the extent that the transition $t_3$ can be used to generate the state in which the book is too late, but nothing in this Petri net represents that this state *should* not have occurred. One might think that a simple solution to add this extra deontic aspect to the standard Petri nets is to add constraints to the transitions. If being too late is undesirable, then why not simply impose a constraint on $t_3$ saying that this transition can only fire if there is no alternative better transition. In this case the Petri net directly fires transition $t_5$ from the initial state. But in this way one forces the Petri net to behave ideally, whereas what we want to model is both ideal and sub-ideal behaviors of agents, and that is quite another thing. Therefore, we propose another solution in this paper.

In this paper we show how Petri net structures can be extended with a preference relation. In the set of all possible executions of a system, intuitively, the preferred ones are those which contain a minimum of sub-ideal behavior. In other words, those executions should be preferred that contain as few sub-ideal transitions as possible. Notice that the set of possible executions of a colored Petri net is equivalent to the set of possible executions of the standard Petri net with the marking given by $M_i = \sum_{j \in C} M_{i,j}$. This is a direct consequence of the fact that we do not refer to the colors of tokens to determine whether a transition is enabled. Hence, instead of analyzing executions of a colored Petri net we can equivalently analyze executions of the standard Petri net. For convenient representation, we do not use colored Petri nets in the remainder of this paper, but we use standard Petri nets instead.

# 3 The extended Petri net formalism

In this section we discuss several ways to extend Petri nets with a preference ordering to model normative behavior. First, we discuss preference orderings on places and show why this extension is not sufficient. Then we discuss preference orderings on transitions. Finally, we discuss as a third option to have preference orderings on executions, and we argue why this is the most appropriate choice.

## 3.1 Preferred places

One way to introduce preferences in Petri nets is to introduce a preference ordering on places.

**Definition 7 ('Preferred places' Petri net)** *A 'preferred places' Petri net structure $N = (P, T, Pre, Post, \geq_P)$ consists of a Petri net structure $N = (P, T, Pre, Post)$ with a partial pre-ordering $\geq_P$ (i.e. reflexive and transitive) on the elements of $P$ such that $p_1 \geq_P p_2$ iff place $p_1$ is preferred to (or equivalent to) $p_2$. We write $p_1 >_P p_2$ iff $p_1 \geq_P p_2$ and not $p_2 \geq_P p_1$, and we write $p_1 \sim_P p_2$ iff $p_1 \geq_P p_2$ and $p_2 \geq_P p_1$.*

The preferences on places can be compared with preferences on states.[3] However, the following example illustrates that this solution is not satisfactory for Petri nets.

**Example 1 (Money)** *Consider the Petri net in Figure 2. The Petri net represents two ways to get hold of an amount of money. The token in $p_1$ can move to place $p_2$ via transition $t_1$ (selling some property) or via transition $t_2$ (stealing the money). In this Petri net, a preference ordering on places cannot distinguish between these two behaviors.*

The previous example shows that a preference ordering on places is sometimes not expressive enough. The place $p_2$ in the money example represents the physical state in which the money is in your possession, which can represent two different deontic states. The deontic state does not depend on the physical state, but how the state was reached. Note that a similar problem appears in the Petri net in Figure 1. The ordering on places $p_1 \sim_P p_3 >_P p_2$ could represent that damaging a book is sub-ideal; hence, it represents that books should not be damaged. However, it is not possible to define an ordering on places such that returning books on time is ideal and returning books too late is sub-ideal (and much too late even more sub-ideal).

[3] Preferences on states are well-known from deontic logic, because several deontic logics have a preferential semantics, for example [Han71, PS94, TvdT96]. Kripke style possible worlds models of these semantics are based on a preference ordering on worlds, and these worlds are usually interpreted as states.

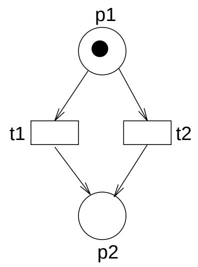


Figure 2: Money example. *Places*: $p_1$: not having money; $p_2$: having money. *Transitions*: $t_1$: selling some property; $t_2$: stealing money.

An alternative way of modeling the Petri net in Example 1 is that we distinguish place $p_2$ in two different places, say $p_{2a}$ and $p_{2b}$. Place $p_{2a}$ then represents the state the money is in after being obtained by selling a property via transition $t_1$, and $p_{2b}$ represents the state the money is in after being obtained by stealing via transition $t_2$. We did not take this approach, because we consider it counterintuitive to represent the very same physical state in two different places only because it can be reached by two different transitions from the same place.

### 3.2 Preferred transitions

Preferences are modeled in Petri nets with priorities [Hac75]. In such a Petri net, a priority is associated with each transition and these priorities can be represented by a partial pre-ordering on transitions.

**Definition 8 ('Preferred transitions' Petri net)** *A 'preferred transitions' Petri net structure $N = (P, T, Pre, Post, \geq_T)$ consists of a Petri net structure $N = (P, T, Pre, Post)$ with a partial pre-ordering $\geq_T$ (i.e. reflexive and transitive) on the elements of $T$ such that $t_1 \geq_T t_2$ iff the priority of $t_1$ is at least as high as the priority of $t_2$. We write $t_1 >_T t_2$ iff $t_1 \geq_T t_2$ and not $t_2 \geq_T t_1$, and we write $t_1 \sim_T t_2$ iff $t_1 \geq_T t_2$ and $t_2 \geq_T t_1$.*

In Petri nets with priorities, the firing rule is changed such that if several transitions can fire, then always one of the most preferred transitions fires.

**Definition 9 (Preferred enabled transition)** *Let $N = (P, T, Pre, Post, \geq_T)$ be a Petri net with marking $M$. A preferred enabled transition is a transition $t \in T$ such that $t$ is enabled in $\langle N, M \rangle$ and for all other enabled transitions $t' \in T$, we have $t \geq_T t'$.*

The following example illustrates that this representation is not satisfactory to model normative behavior in Petri nets, because the notion of choice is not modeled.

**Example 2** *Consider the 'preferred transitions' Petri net $N = (P, T, Pre, Post, \geq_T)$ that consists of the Petri net structure $N = (P, T, Pre, Post)$ of Figure 1, and the ordering $t_2 \sim_T t_5 \sim_T t_6 >_T t_3 \sim_T t_4 >_T t_1$. The ordering represents that the enabled transitions $t_1$ and $t_3$ are sub-ideal. It can easily be shown that in this Petri net the token in place $p_1$ never reaches place $p_2$, because transition $t_5$ is the only preferred enabled transition.*

This notion of preferred transitions makes that the Petri net can only model ideal behaviors, i.e. all executions of the net are ideal. But modeling deontic aspects also means to be able to model sub-ideal behavior, otherwise the notion of the violation of an obligation cannot be modeled in the net. Moreover, when it is impossible to violate a rule it does not make sense to consider such a rule as an obligation, for the same reason physical laws are not considered to be obligations. Hence, the Petri net must have executions that represent even the deontically least desirable behaviors. But such sub-ideal executions are ruled out by the technique of preferred transitions.

Moreover, this solution lacks in expressiveness because it only optimizes local behavior. For example, suppose that an ideal transition forces you to fire a very sub-ideal transition later to reach some goal place, whereas a sub-ideal transition now will prevent this very sub-ideal transition later. A strategy which selects the preferred transition (like Definition 9) is not able to select the latter execution. Hence, it cannot model that *a little lie should be told at this moment to prevent some disaster in the future.* In order to avoid the problem of local optimization, it is much more appropriate to define the preference ordering on executions, as is illustrated in the following section.

### 3.3 Preferred executions

We can derive a preference ordering on executions by partitioning the set of transitions of a Petri net into two subsets, that represent ideal and sub-ideal transitions respectively.[4]

**Definition 10 ('Two transitions' Petri net)** *A 'two transitions' Petri net structure $N = (P, T, Pre, Post, S)$ consists of a Petri net structure $N = (P, T, Pre, Post)$ and a set $S \subseteq T$.*

Intuitively, the set $S$ of a 'two transitions' Petri net contains sub-ideal transitions of $T$. Given this partition in ideal and sub-ideal transitions, we can define a preference ordering on executions of a net that compares sub-ideal transitions of the executions.

**Definition 11 ($\geq_{p_1}$)** *Let $N = (P, T, Pre, Post, S)$ be a 'two transitions' Petri net, $M_1$ and $M_2$ two markings, and $s$ and $s'$ two executions between $M_1$ and $M_2$. Execution $s$ is preferred to execution $s'$, written as $s \geq_{p_1} s'$, iff no sub-ideal transition $t$ occurs more often in $s$ than in $s'$.*

The following example illustrates this preference ordering on executions.

**Example 3** *Let $N = (P, T, Pre, Post, S)$ be a 'two transitions' Petri net, such that $P$, $T$, $Pre$ and $Post$ are given by the Petri net in Figure 1 and the set $S = \{t_1, t_3, t_4\}$. Consider the markings $M_1 = \langle 1, 0, 0 \rangle$ and $M_2 = \langle 0, 0, 1 \rangle$. We have $\langle t_3, t_5 \rangle \geq_{p_1} \langle t_3, t_1, t_6 \rangle$, because the first execution contains less sub-ideal behaviors than the second one. This is intuitively correct, we prefer an execution in which one returns a book one week too late but undamaged to an execution in which one returns the book too late and damaged.*

*However, the executions $\langle t_3, t_5 \rangle$ and $\langle t_1, t_6 \rangle$ are incomparable for $\geq_{p_1}$. This is unintuitive, because one prefers that a book is returned too late to a book that is returned in time with damages (a preference that is usually expressed by a difference in the sanctions associated with the violations). The order relation $\geq_{p_1}$ does not take the differences into account which*

[4] Such a binary distinction exists in the semantics of several deontic logics, most notably in so-called 'standard' deontic logic (SDL).

*can exist between sub-ideal behaviors, because it takes only the number of sub-ideal behaviors into account. However, violations do not have the same seriousness.*

Differences between sub-ideal behaviors can be taken into account by the ordering relation '$\geq_T$' given in Definition 8, which expresses the deontic notion of ideal and varying sub-ideal. We now consider Definition 8 in the context of the usual Petri net firing rule (both ideal and sub-ideal may fire). Given the distinction between ideal and varying sub-ideal transitions, the new problem is how to compare executions. The following definition is an example of a partial pre-ordering on executions derived from the ordering '$\geq_T$' on transitions.

**Definition 12** ($\geq_{p_2}$) *Let $N$ be a 'preferred transitions' Petri net $N = (P, T, Pre, Post, \geq_T)$ as defined in Definition 8, $S = \{t \mid \exists t' \in T : t' >_T t\}$ the set of sub-ideal transitions of $N$, and $s$ and $s'$ two executions between $\langle N, M_1\rangle$ and $\langle N, M_2\rangle$. $s$ is preferred to $s'$ with respect to $\geq_T$, written as $s \geq_{p_2} s'$, iff there exists a function $f$ from the sub-ideal elements of $s$, i.e. $dom(f) = \{s_i \mid s_i \in S\}$, to elements of $s'$ such that:*

1. *if $f(s_i) = s'_j$ then $(s_i \geq_T s'_j)$.*
2. *$\forall s_i, s_j \in dom(f)$ : if $s_i \neq s_j$ then $f(s_i) \neq f(s_j)$*

*Two executions are called equivalent, written as $s \sim_{p_2} s'$, if $s \geq_{p_2} s'$ and $s' \geq_{p_2} s$. Two executions are called incomparable if neither $s \geq_{p_2} s'$ nor $s' \geq_{p_2} s$.*

The previous Definition 12 says that an execution $s$ is preferred to a second one $s'$ if for every instance of a sub-ideal transition in $s$, there is a distinct instance of a transition in $s'$ which is at least as sub-ideal. Restriction 1 on the function $f$ ensures that each instance of a sub-ideal transition in $s$ is *covered* by an instance of a transition in $s'$ that is at least as sub-ideal. Restriction 2 ensures that two instances of sub-ideal transitions in $s$ are not *covered* by the same instance in $s'$. The following example illustrates this ordering on executions.

**Example 4** *Consider the 'preferred transitions' Petri net $N = (P, T, Pre, Post, \geq_T)$ of Example 2 that consists of the Petri net given in Figure 1 and the preference ordering $t_2 \sim_T t_5 \sim_T t_6 >_T t_3 \sim_T t_4 >_T t_1$. The preference ordering represents that returning a book one week too late is preferred to damaging it. For example, the execution $\langle t_3, t_5\rangle$ is preferred to the execution $\langle t_1, t_6\rangle$ for $\geq_T$, because $t_3$ in the first execution can be* covered *by $t_1$ in the second execution.*

*However, also with this definition problems subsist. For example, the executions $\langle t_1, t_2, t_5\rangle$ and $\langle t_1, t_6\rangle$ are equivalent for $\geq_T$. Intuitively this is incorrect, because we want to prefer executions in which one repairs a damaged book to executions in which one does not repair it.*

A solution to the problem mentioned in the previous example without explicitly introducing the notion of repair is illustrated in the following example.

**Example 5** *Reconsider the previous example and let $\geq_T$ be such that that $t_3 \sim_T t_4 >_T t_1$ and $t_2 >_T t_6$. According to the qualitative definition $\langle t_1, t_2, t_5\rangle$ is preferred to $\langle t_1, t_6\rangle$. However, this solution does not seem very intuitive. For example, the executions $\langle t_1, t_2\rangle$ and $\langle t_1\rangle$ are equivalent for $\geq_T$, which seems counterintuitive because the former is preferred. Note that we compare executions which result in a different marking. In this sense, this counterintuitive example differs from previous examples.*[5]

[5] Executions which result in different markings are compared in Section 3.4.1, when we consider preference relations on markings derived from preference relations on executions.

In the previous example, the preferences are related to the deontic notion of contrary-to-duty obligations, as discussed in Section 2. For example, the obligation to repair the damaged book is a CTD obligation. The previous example suggests to treat transition $t_2$ as a particular one. We call it a repair transition of a sub-ideal behavior. A solution which explicitly introduces the notion of repair is given by the following definition. Repair transitions are used to compare executions which are equivalent in the previously defined ordering $\geq_{p_2}$.

**Definition 13** ($\geq_{p_3}$) *Let $N = (P, T, Pre, Post, \geq_T)$ be a 'preferred transitions' Petri net, $S = \{t \mid \exists t' \in T : t' <_T t\}$ and $R \subseteq T$ disjoint sets of sub-ideal and repair transitions of $N$ respectively, $M_1$ and $M_2$ two markings, and $s$ and $s'$ two executions between $M_1$ and $M_2$. $s$ is preferred to $s'$ with respect to $\geq_T$ and $R$, written as $s \geq_{p_3} s'$, iff*

- *either $s >_{p_2} s'$*
- *or $s \sim_{p_2} s'$ and there exists a function $g$ from the repair transitions of $s'$, i.e. $dom(g) = \{s'_i \mid s'_i \in R\}$ such that :*
  1. $g(s'_i) = s_j \rightarrow s'_i = s_j$
  2. $\forall s'_i, s'_j \in dom(g) : s'_i \neq s'_j \rightarrow g(s'_i) \neq g(s'_j)$

Intuitively, if two executions are considered equivalent by ordering $\geq_{p_2}$,then Definition 13 further inspects if one contains more repairs than the other. Restriction 1 imposes that for each repair in $s'$ there exists the same repair in $s$. Restriction 2 of Definition 13 imposes that each instance of a repair transition in $s'$ is mapped to a different instance of transition in $s$. If such a function $g$ exists then $s'$ contains *at most* the same repair transitions as $s$. Obviously, it is also possible to order the repair transitions similar to the ordering on sub-ideal transitions. The following example illustrates the definition of $\geq_{p_3}$.

**Example 6** *Reconsider the previous example and let $t_2$ be a repair transition of the sub-ideal transition $t_1$. We obtain the following intuitive result : $\langle t_1, t_2 \rangle >_{p_3} \langle t_1 \rangle$ since $\langle t_1, t_2 \rangle \sim_{p_2} \langle t_1 \rangle$ and the first execution contains the repair $t_2$ whereas the second one does not. Moreover, we obtain for example that $\langle t_1, t_2, t_1, t_2, t_5 \rangle >_{p_3} \langle t_1, t_2, t_1, t_6 \rangle$.*

### 3.4 Preferred executions - quantitative

In this section, we introduce the penalty function $w$, which expresses how good or bad a transition is. This penalty function associates a weight with every transition.[6] Hence, the ordering defined by the penalty function is total (for every two transitions $t_1$ and $t_2$, we have $t_1 \geq t_2$ or $t_2 \geq t_1$). A positive weight (i.e. a penalty) is assigned to sub-ideal transitions of $T$. This weight will be great if the transition is very sub-ideal. Moreover, the penalty function defines the set of repair transitions of sub-ideal behaviors, because the set contains all transitions with a negative weight. Intuitively, the negative penalties can be considered as rewards for good behavior. Intuitively, a repair transition is a transition that has a negative weight in order to recover from a sub-ideal situation that was brought about by sub-ideal behavior. An extended Petri net is a Petri net with varying sub-ideal and repair transitions.

[6] This definition of the penalty function is very abstract. In Section 4 we will discuss some constraints on this function that make it more realistic from a deontic point of view.

**Definition 14 (Extended Petri net)** *Let $N = (P, T, Pre, Post)$ be a Petri net and $\mathbb{Z}$ the set of integers. An extended Petri net $EN = (P, T, S, R, w, Pre, Post)$ is the extension of $N$ with two disjunct sets $S, R \subseteq T$ that represent sub-ideal and repairing behavior respectively, and a penalty function $w : T \to \mathbb{Z}$, defined as follows :*

1. $t \in T \backslash \{S \cup R\} : w(t) = 0$
2. $t \in S : w(t) \geq 0$
3. $t \in R : w(t) \leq 0$

The penalty function is introduced to facilitate the derivation of preferences between executions from preferences on transitions. In the previous section it was illustrated that this is a non-trivial problem. A simple solution with our penalty function $w$ is that an execution $s$ is preferred to an execution $s'$ iff the sum of the weights of the transitions of $s$ is less than that of the transitions of $s'$. Intuitively, the weights represent penalties for the violations and the preference relation prefers a minimal total sum of penalties. Notice that this definition makes all executions comparable.[7]

**Definition 15 (Preference ordering on executions)** *Let $EN = (P, T, S, R, w, Pre, Post)$ be an extended Petri net, $M_1$ and $M_2$ two markings, $\geq_p$: $E(EN, M_1, M_2) \times E(EN, M_1, M_2)$ a preference relation defined on the set $E(EN, M_1, M_2)$ of the possible executions of $EN$ from the marking $M_1$ to the marking $M_2$, $s, s' \in E(EN, M_1, M_2)$ two executions, and the function $lg(s)$ the length of the tuple $s = \langle s_1, ..., s_n \rangle$ (here equal to $n$). $s$ is preferred to $s'$, written as $s \geq_p s'$ , iff $\sum_{i=1}^{lg(s)} w(s_i) \leq \sum_{j=1}^{lg(s')} w(s'_j)$.*

The preference ordering can be used to determine the preferred executions. A preferred execution is a global optimum, thus we do not have the drawbacks of local optimization (as was observed at the end of Section 3.2).

**Definition 16 (Preferred execution)** *Let $EN = (P, T, S, R, w, Pre, Post)$ be an extended Petri net, $M_1$ and $M_2$ two markings, and $s \in E(EN, M_1, M_2)$ an execution from $M_1$ to $M_2$. The execution $s$ is a preferred execution from $M_1$ to $M_2$ iff $\forall s' \in E(EN, M_1, M_2)$ we have $s \geq_p s'$. We write $EP(EN, M_1, M_2)$ for the set of preferred executions between $M_1$ and $M_2$.*

There is a problem with the previous definition of preferred execution. Consider a family of executions of the form $s = s_1(s_2)^*s_3$, where $s_1, s_2, s_3$ are sequences of transitions, $M_1 \to^s M_2$ and $(s_2)^*$ stands for the repetition of sequence $s_2$ an arbitrary number of times. If $\sum_{i=1}^{lg(s_2)} s_{2,i} < 0$ then there does not exist a preferred execution between $M_1$ and $M_2$, because the execution $s_1(s_2)^{n+1}s_3$ is preferred to $s_1(s_2)^n s_3$ (where $n$ is any natural number). This situation corresponds to a cycle with negative weight in the reachability graph of the Petri net. For the moment, we exclude this case, in the following section we will define conditions for the penalty function $w$ and the structure of an extended Petri net to ensure that no cycle in the reachability graph has a negative weight.

In the following example we use some new graphical notations in addition to the usual notations of the Petri net formalism. A place is represented in the usual way by a circle. If

[7] This is an obvious drawback, but fortunately there are also some obvious refinements of the definition. For example, we can define several types of penalties.

$t_i \in T \backslash \{S \cup R\}$, it is represented in the usual way by a bar (the weight is not represented, because it is always equal to zero). A $t_j \in S$ corresponds to a sub-ideal behavior and is represented by a black bar with a positive integer which represents the weight of the transition. A $t_k \in R$ corresponds to a repair behavior and is represented by a greyed bar with a negative integer which represents the weight of the transition. The following example illustrates the new graphical notation.

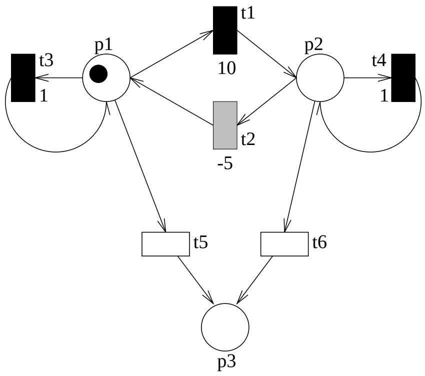


Figure 3: Borrowing books (continued). *Places*: $p_1$: borrowed book; $p_2$: damaged book; $p_3$: returned book. *Transitions*: $t_1$: to damage the book; $t_2$: to repair the book; $t_3$: 1 week too late; $t_4$: 1 week too late; $t_5$: to return the book; $t_6$: to return the book. *Penalty system*: $\langle w(t_1) = 10, w(t_2) = -5, w(t_3) = 1, w(t_4 = 1) \rangle$

**Example 7** *Consider the extended Petri net in Figure 3. It can easily be shown that we obtain the intuitive result* $\langle t_1, t_2, t_5 \rangle >_p \langle t_1, t_6 \rangle$. *Furthermore, we have that* $\langle t_5 \rangle >_p \langle t_1, t_2, t_5 \rangle$. *This states that we prefer that the borrower does not damage the book even if he repairs it.*

### 3.4.1 Preferred reachable states

The preferences on transitions model what ought to be done. Besides these ought-to-do obligations also ought-to-be obligations can be defined, which are preferences on the markings (the states). Preferences on markings can be derived from preferences on transitions and vice versa. In this section we show how preferences on markings can be derived in our extended Petri nets. The preference relation on markings is defined on all reachable markings.

**Definition 17 (Reachable markings)** *Let* $EN = (P, T, S, R, w, Pre, Post)$ *be an extended Petri net with marking* $M$, *and* $T^*$ *the set of all sequences that can be composed from transitions of* $T$. *The set of reachable markings for the marked net* $\langle N, M \rangle$ *is* $R(EN, M) = \{M' \mid \exists s \in T^* : M \rightarrow^s M'\}$.

Given the initial marking $M$, a reachable marking $M_1$ is preferred to marking $M_2$, iff a preferred execution which leads from $M$ to $M_1$, has a weight less than the weight of a preferred execution which leads from $M$ to $M_2$.

**Definition 18 (Preference ordering markings)** *Let* $EN = (P, T, S, R, w, Pre, Post)$ *be an extended Petri net with marking* $M$, $\geq_M$: $R(EN, M) \times R(EN, M)$ *a preference relation defined on the set of the reachable markings of* $EN$, $M_1, M_2 \in R(EN, M)$ *two markings* *and $lg(s)$ the length of execution $s$. $M_1$ is preferred to $M_2$, written as $M_1 \geq_M M_2$, iff $\sum_{i=1}^{lg(s_1)} w(s_{1,i}) \leq \sum_{j=1}^{lg(s_2)} w(s_{2,j})$, where $s_1 \in EP(EN, M, M_1)$ and $s_2 \in EP(EN, M, M_2)$.*

The following example illustrates the preference ordering on markings.

**Example 8** *Consider our running example with the initial marking $\langle 1, 0, 0\rangle$. The marking $\langle 0, 0, 1\rangle$ is preferred to the marking $\langle 0, 1, 0\rangle$, because the preferred execution from $\langle 1, 0, 0\rangle$ to $\langle 0, 0, 1\rangle$ is $\langle t_5\rangle$ and its penalty weight is 0, whereas the preferred execution from $\langle 1, 0, 0\rangle$ to $\langle 0, 1, 0\rangle$ is $t_1$ and its penalty weight is 10.*

### 3.5 Preferred markings

In the previous Sections 3.3 and 3.4, we have argued that different types of behavior should be modeled by a preference ordering on executions, which can be derived from a preference ordering on transitions. In particular, we have argued that such an ordering on executions is preferred over a preference ordering on places (described in Section 3.1) or a preference ordering on transitions (described in Section 3.2). When we use colored Petri nets, there is another possibility which we discuss in this section: a preference ordering on colored markings. In this approach, the colors of the tokens represent the deontic status of the token. The following example illustrates such a preference ordering by formalizing penalties as colors of tokens.

**Example 9** *Let $CN = \langle P, T, Pre, Post, C, F\rangle$ be a colored Petri net that consists of the Petri net in Figure 1, the set of colors $C$ is the set of natural numbers and the transition function $F$ is such that the color in increased by one by transitions $t_3$ and $t_4$, increased by 10 by transition $t_1$ and decreased by 5 by transition $t_2$. We can compare the the color of the token with penalties of executions (sum of penalties of the transitions in the execution) of the extended Petri net in Figure 3. Obviously, given a single token in place $p_1$ with color 0, the penalty of an execution is the color of the token at the end of the execution. In case of multiple tokens, the penalty of an execution is the sum of the colors of the tokens after the execution.*

The previous example illustrates that the ideas defined in the previous Section 3.4 can be redefined in terms of a preference ordering on colored markings of Petri nets. Similarly, the qualitative ordering used in Section 3.3 can be redefined in terms of a preference ordering on colored markings, when the color represents the whole trace of the token, instead of only the penalties.[8] Whether we represent the deontic status of a procedure by preferences on executions or preferences on colored markings seems to be a modeling decision.

## 4 Preferences and Deontic Notions

In the previous section we have shown that deontic aspects can be modeled by preferences. However, not every preference relation represents deontic aspects. In this section we further discuss the relation between preferences and deontic aspects, and we give a few possible conditions on the preference ordering. For simplicity we restrict ourselves to the quantitative approach of Section 3.4.

[8] Note that an ordering on markings is slightly more expressive, because we can discriminate between the penalties of individual tokens. For example, in Figure 1 with an initial marking of two tokens in place $p_1$, we can discriminate between one book being two weeks to late, and two books being each one week too late.

## 4.1 Induced obligations

The penalty system that is introduced in the extended Petri net induces obligations, permissions and prohibitions. An execution fulfills the induced obligations when it is in the set of preferred executions.

**Definition 19 (Fulfilled $w$-induced obligations)** *Let $EN = (P, T, S, R, w, Pre, Post)$ be an extended Petri net, $M_1, M_2$ two markings, $s$ an execution such that $M_1 \rightarrow^s M_2$. The execution $s$ fulfills the $w$-induced obligations iff $s \in EP(EN, M_1, M_2)$ (the set of preferred executions from $M_1$ to $M_2$ in $EN$).*

Given the previous definition of executions that fulfill obligations, we define obliged, permitted and prohibited transitions. In the following $s_0$ denotes the first transition of the executions $s$.

**Definition 20 ($w$-induced Obligation)** *Let $EN = (P, T, S, R, w, Pre, Post)$ be an extended Petri net, $M_1, M_2$ two markings such that $\exists s \in T^* : M_1 \rightarrow^s M_2$, a transition $t$ is a $w$-induced obligation in $M_1$ to reach $M_2$ noted $obliged(t, EN, M_1, M_2)$ iff $\forall s \in E(EN, M_1, M_2)$ we have $s_0 = t$.*

**Definition 21 ($w$-induced Permission)** *Let $EN = (P, T, S, R, w, Pre, Post)$ be an extended Petri net, $M_1, M_2$ two markings such that $\exists s \in T^* : M_1 \rightarrow^s M_2$, a transition $t$ is a $w$-induced permission in $M_1$ to reach $M_2$ noted $permitted(t, EN, M_1, M_2)$ iff $\exists s \in E(EN, M_1, M_2)$ such that $s_0 = t$.*

**Definition 22 ($w$-induced Prohibition)** *Let $EN = (P, T, S, R, w, Pre, Post)$ be an extended Petri net, $M_1, M_2$ two markings such that $\exists s \in T^* : M_1 \rightarrow^s M_2$, a transition $t$ is a $w$-induced prohibition in $M_1$ to reach $M_2$ noted $forbidden(t, EN, M_1, M_2)$ iff $\neg\exists s \in E(EN, M_1, M_2)$ such that $s_0 = t$.*

Note that when transition $t$ is *obliged* or *permitted* in a marking $M$ then $t$ is *enabled* in this marking. The following proposition gives the relation between the deontic notions.

**Proposition 1** *The relation between obliged, permitted and forbidden is as follows.*

$$obliged(t, EN, M_1, M_2) \Rightarrow permitted(t, EN, M_1, M_2)$$

$$permitted(t, EN, M_1, M_2) \Leftrightarrow \neg forbidden(t, EN, M_1, M_2)$$

The following example illustrates these definitions.

**Example 10** *In marking $\langle 1, 0, 0\rangle$ the only sequence to reach $\langle 0, 0, 1\rangle$ which fulfills the $w$-induced obligations is $\langle t_5\rangle$. In that marking we have $obliged(t_5, EN, \langle 1, 0, 0\rangle, \langle 0, 0, 1\rangle)$. In the same marking, if the obligation to fire $t_5$ is violated by firing $t_1$ (damaging the book), we obtain the marking $\langle 0, 1, 0\rangle$ where the transition $t_2$ is $w$-obliged. This is sound as firing the transition $t_2$ represents a CTD-obligation after firing $t_1$.*

## 4.2 Structure of the net

There is no formal relation between a sub-ideal transition and its repair transitions in the extended Petri nets. This may be a drawback, because it might be possible to repair sub-ideal transitions with unrelated repairs. For example, in a library example it might be possible to repair the violation of being to late by repairing a book. Fortunately, this is not possible in the Petri net given in Figure 1, because transition $t_2$ (repairing a book) has to be preceded by transition $t_1$ (damaging a book). In general, the structure of the net will impose several restrictions on behaviors which ensure that the preferences can be interpreted as a formalization of different types of behavior.

This observation raises the question which extra constraints can be imposed on the nets in order to get certain desirable properties that make the penalty function more realistic from a deontic point of view. For example, it seems reasonable that an extended Petri net should satisfy the property that *a repair transition should always be preceded in an execution by a sub-ideal transition.* If we make the simplification assumption that a sub-ideal transition has at most one repair transition associated, the following condition on the structure of an extended Petri net is an example of a *sufficient condition* to obtain the property.

**Condition 1** *Let $EN = (P, T, S, R, w, Pre, Post)$ be an extended Petri net, $M_0$ the initial marking of EN, and $Rep : S \times R$ a relation. For every $s$ and $r$ such that $Rep(s, r)$, there is a $p \in P$ such that:*

1. $M_0(p) = 0$
2. $Post(p, s) - Pre(p, s) = 1$
3. $\forall t \in T \backslash \{s\} : Post(p, t) - Pre(p, t) \leq 0$
4. $Post(p, r) - Pre(p, r) = -1$

In this condition $Rep(s, r)$ expresses that $r$ is the repair transition of the sub-ideal transition $s$, and we assume that some place $p$ witnesses sub-ideal (and corresponding repair) transitions. Remember that we make the simplification assumption that a sub-ideal transition has at most one repair transition associated. Item 1 of Condition 1 imposes that the place $p$ is empty initially. Item 2 expresses the fact that when the sub-ideal behavior $s$ is performed then a token is added to the place $p$. In item 3 it is expressed that the only transition that can add tokens to the place $p$ is the sub-ideal transition $s$. Finally, item 4 says that firing the repair transition $r$ removes a token of the place $p$.

Another example of a property is that *it is better not to do a sub-ideal behavior than doing it first and repairing it afterwards.* A simple way to obtain this property is to impose on the penalty function the following constraint.

**Condition 2** *Let $EN = (P, T, S, R, w, Pre, Post)$ be an extended Petri net, $Rep : S \times R$ a relation such that $Rep(s, r)$ expresses that $r$ is the repair transition of the sub-ideal transition $s$ then $-w(r) < w(s)$.*

**Example 11** *In the extended Petri net of our running example (see Figure 3), we have that $Rep(t_1, t_2)$ is true. The place $p_2$ fulfills the 4 requirements of Condition 1 and we have that $-w(t_2) = 5$ and $w(t_1) = 10$ which fulfills Condition 2 since $5 < 10$.*

The following proposition states that there always exists a preferred execution.

**Proposition 2** *Let $EN = (P, T, S, R, w, Pre, Post)$ be an extended Petri net which fulfills Condition 1 and Condition 2, in that case, if a marking $M_2$ is reachable from a marking $M_1$ then there always exists a preferred execution between these two markings.*

Obviously, there are several further conditions which can be defined on the preference relation. For example, in the penalty system we can define a set of possible initial and goal markings, and then we can add the constraint that there has to be at least one execution from each initial marking to each goal marking with penalty 0. Such a condition expresses that ideal (i.e. violation free) behavior is possible, a well-known principle accepted by many deontic logics.

# 5 Conclusions

In this article we have shown how to represent ideal and sub-ideal deontic behavior in Petri nets by extending these nets with a preference relations. For example, a preference relation was defined by introducing a penalty function on transitions of the Petri net, from which a preference ordering on possible executions was derived (which represent its procedural semantics). We have also shown that by ordering the transitions of the net, we can define a preference relation on the reachable states of an extended marked Petri net. Algorithms to compute the preference relation in Petri nets, based on graph-search-path techniques, are given in [Ras94].

The Petri net formalism style is operative in contrast with the declaration style of logic formalisms. It has been shown [LR95] how the Petri net formalism can be extended (for example with temporal logic) to be used as a description (specification) language. In that paper, an operative solution is proposed to model deontic notions in Petri nets. An alternative approach is to define a more declarative way to express deontic aspects. For example, a net could be annotated with logic formulae such as : $Fired(t_1) \Rightarrow I(\Diamond(Fired(t_2)))$, which would express on the Petri net of figure 1 that *'if one damages a book (firing $t_1$), idealy ($I$) one should repair it (firing $t_2$)'*.

# 6 Acknowledgement

Thanks to Patrick van der Laag for reading a previous version of this paper.

# References

[BLWW95] R.W.H. Bons, R.M. Lee, R.W. Wagenaar, and C.D. Wrigley. Modeling inter-organizational trade procedures using documentary Petri nets. In *Proceedings of the Hawaii International Conference on System Sciences (HICSS'95)*, 1995.

[Chi63] R.M. Chisholm. Contrary-to-duty imperatives and deontic logic. *Analysis*, 24:33–36, 1963.

[For84] J.W. Forester. Gentle murder, or the adverbial Samaritan. *Journal of Philosophy*, 19:75–93, 1984.

[Gen86] H.J. Genrich. *Predicate Transition Nets*, volume Computer Sciences, vol. 254. Springer, Berlin, 1986.

[Hac75] M. Hack. Petri net languages. Technical report, MIT, Computations Structures group, 1975.

[Han71] B. Hansson. An analysis of some deontic logics. In *Deontic Logic: Introductionary and Systematic Readings*, pages 121–147. D. Reidel Publishing Company, Dordrecht, Holland, 1971.

[JS92] A.J.I. Jones and M. Sergot. Deontic logic in the representation of law: towards a methodology. *Artificial Intelligence and Law*, 1, 1992.

[Lee91] R.M. Lee. CASE/EDI: Edi modeling, user documentation. Technical report, EURIDIS, Erasmus University Rotterdam, 1991.

[Lee92] R.M. Lee. Dynamic modeling of documentary procedures: A case for edi. Technical Report 92.06.01, EURIDIS, Erasmus University Rotterdam, 1992.

[LR95] B. Ligny and J.-F. Raskin. Modeling distributed systems with petri nets and temporal logic. master's thesis. Technical report, 1995.

[MP92] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurent Systems*. Springer Verlag, 1992.

[Pet81] J.L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, 1981.

[PS94] H. Prakken and M.J. Sergot. Contrary-to-duty imperatives, defeasibility and violability. In *Proceedings of the Second Workshop on Deontic Logic in Computer Science*, Oslo, 1994.

[Ras94] J.-F. Raskin. Modeling sub-ideal deontic states in petri nets. Technical report, EURIDIS, Erasmus University Rotterdam, 1994.

[TvdT95] Y.-H. Tan and L.W.N. van der Torre. Why defeasible deontic logic needs a multi preference semantics. In *Proceedings of the ECSQARU'95*. Springer Verlag, 1995.

[TvdT96] Y.H. Tan and L.W.N. van der Torre. How to combine ordering and minimizing in a deontic logic based on preferences. In *Proceedings of the third workshop on deontic logic in computer science (Deon'96)*. Springer Verlag, 1996.

[vdA92] W.N.P. van der Aalst. *Timed Colored Petri Nets and their Application to Logistics*. PhD thesis, 1992.