# Inconsistent state analysis of a network receiver with colored Petri nets

Mohamed Bouali, Pavol Barger, Walter Schön

# Inconsistent State Analysis of a Network Receiver with Colored Petri Nets

Mohamed BOUALI, Pavol BARGER and Walter SCHON
*Heudiasyc Laboratory, CNRS UMR 6599*
*Université de Technologie de Compiègne*
*Compiègne, France*
{*mohamed.bouali,pavol.barger,walter.schon*}*@hds.utc.fr*

## Abstract

*This paper deals with a new method to study dependability of distributed systems using Colored Petri Nets (CPN) which are a powerful, recognized and intuitive modelling tool. They allow a precise representation of the studied systems. The CPN analysis gives information about static and dynamic behavior of the modelled system and can be used to study questions concerning its dependability. This paper develops a new method of CPN analysis called the Backward reachability. It provides information about different ways of reaching a particular CPN marking that represent, for example, a failure state or a transilient fault. This analysis is performed on an inverse CPN which is obtained by transforming original CPN structure. The illustrative case is the study of a communication protocol composed of three parts: the sender, the receiver ans the network. The Backward reachability analysis done on this system starts by supposing an inconsistent sate in the receiver and looks for the corresponding sender state. The main advantage of this method is that it determines the sequence leading from the initial to the final state independently on the final marking value.*

## 1. Introduction

System dependability is an important research issue especially if applied to critical domain. Systems are verified through tools that check the compliance of their properties with design specifications. Formal methods provide an interesting way to study and develop verification tools thanks to their precise characterization of the modelled system. The ever increasing complexity of new systems requires new tools. It is in this context that the presented work applies a formal method approach to system modelling and analysis.

A concrete application of formal methods is the detection of transient and inconsistent states. These ones can appear randomly in a system and cause its crash. Communication protocols are a commun example of a system in which inconsistent states can easily appear. They are mostly composed of three parts: a *sender* and a *receiver* communicating through a *network*. The fault can be caused by the communication medium, intrusion, data processing, etc. In such system, a dependability criteria can be the non existence of the undesired state. To do this, this study uses Colored Petri Nets as formal tool and develops a certain type of its analysis.

Petri Nets (PN) [8], and especially, Colored Petri Nets(CPN) [3], are a powerful and recognized modelling tool. They are endowed with a big expressiveness and allow to represent the two aspects of a system : static thanks to the PN structure and dynamic thanks to the token evolution. The PN analysis can be done in several manners: exhaustive reachable state space enumeration, simulation, structural analysis, etc. These methods allow to study request/action effects on the model behavior. The exhaustive reachable state space enumeration gives correct results but suffers from the combinatorial explosion. The Monte Carlo simulation [7] is a robust method but although results are statistically correct, they are bound with a known error range. To avoid drawbacks of these methods, we propose to use the structural analysis using directly the model itself.

Usually, the performed analysis is the forward one. That means, by knowing an initial state, possible final states are calculated. This is particularly adapted for the studies of performances and the quality of service (QoS). But, it is not adapted to find sources of some particular final state which can be, for example, a failure state. In such analysis, all possible initial configurations must be studied to find those leading to the considered failure state. This is why, in the case of ordinary Petri Nets, [4] interested in the reachability between two markings studied under two dual manners: the forward reachability and the backward reachability. The first case consists in building state successors one by one, starting with the initial marking $M_0$ and ending with the final marking $M_f$. Thus, considering $M_0$ as the present state, $M_f$ is considered as the future state. In the backward reachability, $M_f$ constitutes the present state and $M_0$ is regarded as the past state. The general idea is to build, from the present state, the predecessor states until the reach of the past state which is logically the source of the present marking. Generally speaking, the forward reachability allows to underline the marking paths which lead to the final marking and those that lead to other markings than the final marking. Likewise, the backward reachability identifies the paths which lead from the initial marking to the final marking, and also those who lead to the final marking and do not start with the initial marking. To perform the backward structural analysis, [4]

chose a method based on inverse Petri Net (inverse PN). This method is defined and applicable for ordinary PNs but not for Colored PNs.

[1] proposes a method, based on [6], for safety analysis using backward reachability of CPN. This method separates token values from transition firing conditions. Possible tokens values are obtained by applying a variation of the CPN fondamental evolution equation allowing the backward evolution of the marking. Firing conditions are obtained by performing logical unification of the possible values with arc expressions. This process is not always possible that is why the method introduced the *don't care condition*. It allows to split places of the CPN into two distinct sets. The first set contains places whose initial marking is known. The second contains places whose initial marking is not known and allows a certain liberty about unknown token values. This mechanism is efficient but it truncates some conditions of the backward firing transition and so hides some firing information.

This paper is organized as follows : The section 2 gives definitions of Petri Nets and Colored Petri Nets. The section 3 presents the inversion of Ordinary Petri Nets. It is followed, in the section 4, by the details of our approach and, in section 6, a presentation of the case study. The paper ends by the conclusion and an outline of future perspectives.

## 2. Definitions

### 2.1. Petri Nets

A Petri Net [8], called also Place/Transition Net, is a directed bipartite graph defined by the 4-tuple $(P, T, Pre, Post)$, where:

- $P$ is a finite set of places,
- $T$ is a finite set of transitions,
- $P \cap T = \emptyset$,
- $Pre$ is the backward incidence application,
- $Post$ is the forward incidence application.

### 2.2. Colored Petri Nets

The notation of Colored Petri Net (CPN) [3] introduces the notion of token types, namely tokens are differentiated by colors, which may be arbitrary data values. Each place has an associated type determining the kind of data that the place may contain. A non-hierarchical CPN is defined by the 9-tuple $(\Sigma, P, T, A, N, C, G, E, I)$ where :

- $\Sigma$ is a finite set on non-empty types,
- $P$ is a finite set of *places*,
- $T$ is a finite set of *transitions*,
- $A$ is a finite set of *arcs* such that: $P \cap T = P \cap A = T \cap A = \emptyset$,
- $N$ is a node function. It is defined from $A$ into $P \times T \cup T \times P$,
- $C$ is a *color* function. It is defined from $P$ into $\Sigma$,
- $G$ is a *guard* function. It is defined from $T$ into expressions such that:

$$\forall t \in T : [Type(G(t)) = B \wedge Type(Var(G(t))) \subseteq \Sigma],$$

- $E$ is an *arc expression* function. It is defined from $A$ into expressions such that:

$$\forall a \in A : [Type(E(a)) = C(p(a))_{MS} \wedge Type(Var(E(a))) \subseteq \Sigma]$$

  Where $p(a)$ is the place of $N(a)$,
- $I$ is an *initialization* function. It is defined from $P$ into closed expressions such that:

$$\forall p \in P : [Type(I(p)) = C(p)_{MS}]$$

For practical reasons, we write $E$ in a split form $Pre$ and $Post$ (as used in [2]) such that $Pre$ (resp. $Post$) is the backward (resp. forward) incidence application. It is defined as $E$ where $p(a)$ is the place of a part of $N(a)$ defined from $A$ to $P \times T$ (resp. $T \times P$).

Fig.1 shows an example of CPN designed with the CPNtools[1]. It contains a set of four places $\{p1, p2, p3, p4\}$ and a set of two transitions $\{t1, t2\}$. Each place is declared from some defined type (called *colset*). Two types are defined: *INT* and *STRING*. The transition $t1$ is associated to the guard $[X > 10]$. That means, only tokens which values satisfy this guard can be fired through $t1$. We note also the arc expressions. $X$ and $Y$ are declared $INT$ variables, $ExFun$ is a function transforming the $X$ value and "$a$" is a $STRING$ constant.

1. http://wiki.daimi.au.dk/cpntools/cpntools.wiki

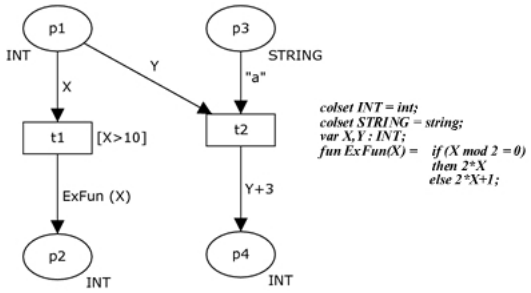



Figure 1. An example of CPN

## 3. inverse Petri Net

In order to perform the backward reachability analysis, the inverse Petri Net (inverse PN) has to be constructed as follows:

Considering a PN $R$ such as $R = (P, T, Pre, Post)$, the inverse PN (noted $R^{-1}$) is defined by $R^{-1} = (P', T', Pre', Post')$ [4] where: $P' = P$ and $T' = T$.

The applications $Pre'$ and $Post'$ are defined as :

$$\forall p \in P, \forall t \in T \begin{cases} Pre'(p,t) = Post(p,t) \\ Post'(p,t) = Pre(p,t) \end{cases}$$

This definition cannot be directly applied to CPN. The definition of $R^{-1}$ (in particular $Pre'$ and $Post'$) is not sufficient as soon as arc expressions differ from the identity function. In addition to this, the backward reachability uses some mechanisms like the *"marking enhancement"* [4]. These mechanisms are efficient in the case of PN but they are incompatible with CPN because of the token value evolution due to arcs valuation. Finally, both PNs $R$ and $R^{-1}$ keep the same structures even though some reductions or transformation might be able to make analysis more efficient.

## 4. Approach presentation

As defined in PNs, backward reachability in CPNs is a dual concept of forward reachability. That is, if a marking $M_f$ is reachable from $M_0$, we say that $M_0$ is backward reachable from $M_f$. By backward reachability, we mean that $M_0$ is a cause or source of $M_f$. For example, In Fig.1 consider a marking $M_f = \langle 8 \rangle.p4$[2]. $M_f$ is reachable from $M_0 = \langle 5 \rangle.p1 + \langle "a" \rangle p3$ and we say that $M_0$ is backward reachable from $M_f$.

Our contribution concerns firstly the generalization of the backward reachability from the PN class to the CPN class and secondly the development of adapted transformation rules. The generalization implies a definition of an inverse CPN which is obtained by performing the proposed transformations on the original CPN.

### 4.1. CPN transformation for the definition of an inverse CPN

Let $R = (\Sigma, P, T, A, N, C, G, E, I)$ be a CPN and let $R' = (\Sigma', P', T', A', N', C', G', E', I')$ be the inverse CPN of $R$. To define $R'$, we perform some transformations on $R$. These transformations are directly dependent on the CPN structure. Consequently, we have to define a transformation for each structure case studied. Nevertheless, the most common transformations in representative cases are presented in this section. Note that, in the remaining paper, $E$ is written in its split form $(Pre, Post)$. So, we write also $E' = (Pre', Post')$.

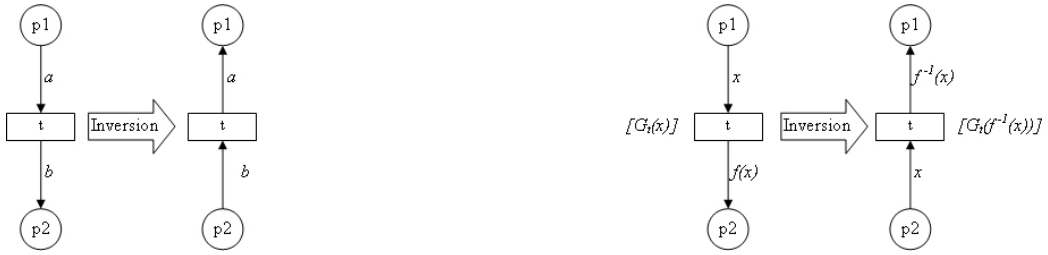
**4.1.1. Basic transformations.** Fig.2.a shows a trivial case of a CPN inversion. Arcs (input and output) are marked with constants $a, b$. In this case, it is enough to generalize the definition of the inverse PN to CPN obtaining :

$$\begin{cases} Pre'(p2,t) = Post(p2,t) \\ Post'(p1,t) = Pre(p1,t) \end{cases}$$

Note that, in the original CPN, $M_0 = \langle a \rangle.p1$ gives $M_1 = \langle b \rangle.p2$ by firing $t$. In the inverse CPN $M_1 = \langle b \rangle.p2$ gives $M_0 = \langle a \rangle.p1$ by firing $t$. Thus, we performed the backward reachability using the inverse CPN.

Fig.2.b shows the case where the input arc is marked with a variable $x$, the output arc is marked by a function $f(x)$ and a guard $G(x)$ can be associated to the transition $t$. Generalized application of the rule shown before (inverting

2. The formula $M_i = \langle V1 \rangle.p1 + \langle V2 \rangle.p2$ means the marking $M_i$ contains token in $p1$ whose value is $V1$ and token in $p2$ whose value is $V2$ (notation taken from [5])




-a- -b-

Figure 2. Basic transformations for CPN inversion

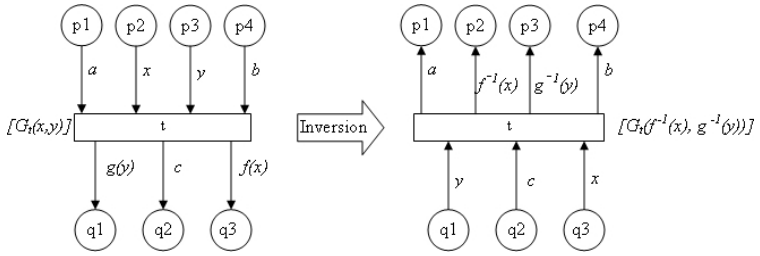



Figure 3. Mixed transformations for CPN inversion

arcs directions) can lead to the construction of an incorrect net. In this example, the input arc would be marked with a function while the output arc would be marked with the variable of this function. That leads to the impossibility for the function evaluation. This is why we suggest marking the input arc by the variable, the output arc by the function $f^{-1}$ and update the guard to express new constraint associated to the transition. This assumes the necessity to know whether function $f$ is reversible. If yes, it is necessary to define its inverse noted $f^{-1}$. This transformation gives :

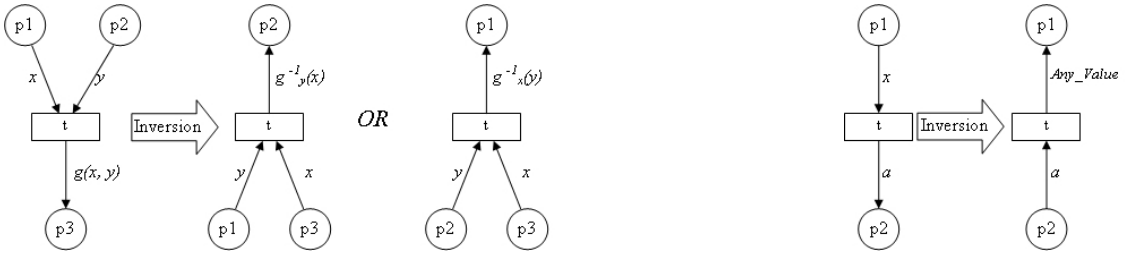
$$\begin{cases} Pre'(p2,t) = Pre^{-1}(p1,t) \\ Post'(p1,t) = Post^{-1}(p2,t) \\ G'_t(Pre'(p2,t)) = G_t(Post^{-1}(p2,t)) \end{cases}$$

$Pre^{-1}(p1,t)$ denotes transformation of the arc $Pre(p1,t)$ which is marked with a new variable defined on $f^{-1}$ domain. $Post^{-1}(p2,t)$ denotes transformation of the arc $Post(p2,t)$ which is marked by $f^{-1}$ (the opposite reverse of $Post(p2,t)$ marking function).

**4.1.2. Mixed transformations.** Fig.3 shows a mixed case where some input arcs are marked with variables $\{x,y\}$ and other arcs by constants $\{a,b\}$. Output arcs are marked with constants $\{c\}$ and reversible functions $\{f,g\}$. This CPN inversion is a mix (generalization) of basic transformations. For arcs marked with a constant, it is sufficient to generalize the rule applied on PNs (changing arcs direction). For the remaining arcs, we have to apply the rule illustrated in Fig.2.b. So, we have to associate each variable to its function in order to respect origine places of arcs marked by variables and sink places which receive resulting tokens. The following constraint has to be verified in this case: each variable is used by one and only one function. If not true, see the *Parametric transformation* subsection below. The algorithm describing the mixed transformation can be written as follow :

- $Pre'(qj,t) = Post(qj,t)$, $j = 1 \ldots m$ if $Post(qj,t)$ is marked with a constant,
- $Post'(pi,t) = Pre(pi,t)$, $i = 1 \ldots n$ if $Pre(pi,t)$ is marked with a constant,
- $Pre'(qj,t) = Pre^{-1}(pi,t)$, $i = 1 \ldots n$, $j = 1 \ldots m$ if $Pre(pi,t)$ is marked with a variable and this variable is associated to the function $Post(qj,t)$,
- $Post'(pi,t) = Post^{-1}(qj,t)$ ,$i = 1 \ldots n$, $j = 1 \ldots m$ if $Post(qj,t)$ is marked with a function and this function is associated to the variable $Pre(pi,t)$,
- $G'_t(Pre'(qj,t)) = G_t(Post^{-1}(qj,t))$ ,$i = 1 \ldots n$, $j = 1 \ldots m$ if $Post(qj,t)$ is marked with a function and this function is associated to the variable $Pre(pi,t)$.

**4.1.3. Parametric transformations.** Some CPN structures can't be reversed to get deterministic markings in the backward reachability. The reason is that the inversion process could be assimilated to mathematic operations whose solutions may be intervals. The CPN inversion, in these cases, is *parametric*. That means, some additional information,




-a- -b-

Figure 4. Parametric transformations for CPN inversion

like color sets, are needed. Two examples follow : the first treats solution of multivariable equations, the second treats input variables which are not associated with output functions.

Fig.4-a- shows a case where output arc is marked by a multivariable function. In this case, the CPN is not directly reversible. The values of variables $\{x, y\}$ can't be deduced by knowing only the value of $g(x, y)$. The *partial (parametric)* solution proposed consists in finding values of one variable knowing other variable values and the function result. As illustrated in Fig.4-a-, firing $t$ (in inverse CPN) requires tokens either in $\{p2,p3\}$ or in $\{p1,p3\}$. Functions $g_x^{-1}(y)$ and $g_y^{-1}(x)$ are *partial inverses* of $g(x, y)$. An example of this transformation is given in the case study.

Fig.4-b- shows a case of variable which is not associated with a function. We note that input arc expression is a variable and output arc (or arcs) expression is constant. This CPN inversion is similar to the case of Fig.2.a, except the arc $Post'(p1, t)$ which is marked by a *parametric* expression noted $'Any\_Value'$. It means that this arc expression can take all values out of $t$ in the place $p1$ color set.

**4.1.4. Parallelism transformations.** The term *'parallelism'* means existence of a shared variable. Fig.5 shows the case where the same variable is used by more than one function (two functions in this case). To inverse this CPN, we have to calculate the inverse of only one function using the shared variable (that supposes existence of, at least, one reversible function). Let $f$ be a reversible function. In the original CPN (shown on the left in Fig.5), transition $t$ firing produces two tokens (in $p2$ and $p3$). Each token value results from a function applied to the shared variable : function $f$ to the token in $p2$ and function $h$ to the token in $p3$. The inverse CPN must produce a token in $p1$ by firing $t$ whose preconditions are $p2$ and $p3$. But it is not enough to have tokens in $p2$ and $p3$ to fire $t$ because token values (in $p2$ and $p3$) must have coherent values towards applied functions $f$ and $h$. For this reason, we define a *guard* associated to transition $t$. It checks that the value of $f^{-1}(x)$ applied to $h$ gives as the result the value of $y$ (token initially in $p3$). If the guard value is *True*, $t$ will be fired, then the initial marking of $p1$ will be found. If the guard value is *False*, $t$ will not be fired.

To illustrate this transformation, let extend the example of Fig.5 where $f(x) = x + 4$, $h(x) = (x > 10)$ and initial marking $M_0 = \langle 5\rangle.p1$ The evolution of this marking is :

$$M_0 = \langle 5\rangle.p1$$
$$t \downarrow$$
$$M_f = \langle 9\rangle.p2 + \langle False\rangle.p3$$

The first test consists to apply backward reachability to $M_f$ on the inverse CPN (Fig.5). The expected result is to determine the initial marking $M_0$.

$$M_f = \langle 9\rangle.p2 + \langle False\rangle.p3$$
$$t \downarrow$$
$$M_0 = \langle 5\rangle.p1$$

Note that the guard gives $'True'$. It is obtained by applying the formula $[y == h(f^{-1}(x))]$ which is $['False' == (5 > 10)]$.

The second test consists to apply backward reachability to an unreachable final marking $M'_f$ on the inverse CPN of the figure. The expected result is to find an empty set of initial markings $M_0$.

$$M'_f = \langle 9\rangle.p2 + \langle True\rangle.p3$$
$$t \downarrow$$
$$\{\}$$

$\{\}$ stands for impossible firing. Note that the guard gives $'False'$. It is obtained by applying the formula $[y == h(f^{-1}(x))]$ which is $['True' == (5 > 10)]$.

The algorithm describing the parallelism transformation can be written as follows:
Let $R$ be a CPN containing a variable $x$ ($Pre(p, t) = x$) which is shared by $n$ functions $f_1, f_2, \ldots, f_n$ ($Post(qi, t) = f_i(x), i = 1 \ldots n$) with $f_1$ inversible. The inverse CPN is defined by :

- $Pre'(q1, t) = Pre^{-1}(p, t)$,
- $Pre'(qi, t) = Id_i, i = 2 \ldots n$,

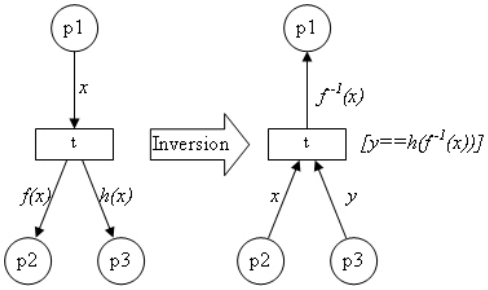



Figure 5. Parallelism transformation for CPN inversion

- $Post'(p,t) = Post^{-1}(q1,t)$,
- $G'_t = \bigwedge_n^{i=2}[Pre'(qi,t) == f_i(f_1^{-1}(Pre'(q1,t)))]$.

## 5. Supplementary mechanisms

The rules presented above allow the inversion of the CPN structure. In order analyse the inversed dynamic behavior two complementary principles are state here.

### 5.1. Marking Enhancement

The marking enhancement is a mechanism allowing to complete the information about a model by adding assumptions about the modelled system state (additional tokens). It is done when constructing successor states. Sometimes, the immediate successor of a given state (which is equivalent to fire a transition) can only be constructed with help of added tokens with appropriate values in certain places. The marking enhancement interpretation consists in assuming that the system is in some given state allowing it to operate. This mechanism has to be performed if only a part of the token distribution is known.

### 5.2. Potentially Valid Transition

In a marked CPN, a transition is potentially valid is it has, at least, 1.) one precondition place marked by a token whose value is compatible with the transition firing (i.e. token compatible with arc expression and guard) ; and 2.) one precondition place which is not marked by a token whose value is compatible with the transition firing.

## 6. Case study

This section shows a practical using of the backward reachability (through an inverse CPN) presented in previous paragraphs. The application example is inspired from *"Simple Protocol"* [3]. It describes a communication protocol with an identified acknowledgement.

The basic question studied here is whether the receiver could possibly receive a hypothetical set of frames.

The CPN model of the protocol (stop-and-wait) is shown in Fig.6. It consists of three parts. The $Sender$ part has two transitions which can *Send Packets* and *Receive Acknowledgments*. The $Network$ part has two transitions which can *Transmit Packets* and *Transmit Acknowledgments*. Finally, the $Receiver$ part is modelled by a single transition which can *Receive Packets* and *send acknowledgments*. The interface between the $Sender$ and the $Network$ consists of places $A$ and $D$, while the interface between the $Network$ and the $Receiver$ consists of places $B$ and $C$. The packets to be sent are placed in place $Send$. Each token in this place contains a packet represented by its sequence number. The place $Next\_Send$ contains the number of the next packet to be sent and is updated each time when the correct acknowledgment is received. All correctly received messages are stored in a buffer represented as place $Receiver$. Place $Next\_Rec$ contains the number of the next packet to be received and is updated each time a packet is successfully received.

The normal working of this system would not allow any doublets (meaning the correct reception of two identical frames). Thus for example the marking with 3 following packets in place $Receiver$ where the first is identified as 1, the two others frames equal to each other and identified as 2 represent and inconsistent state. In the following this marking will be noted as $\langle 1+2+2\rangle.Receiver$.

3. http://wiki.daimi.au.dk

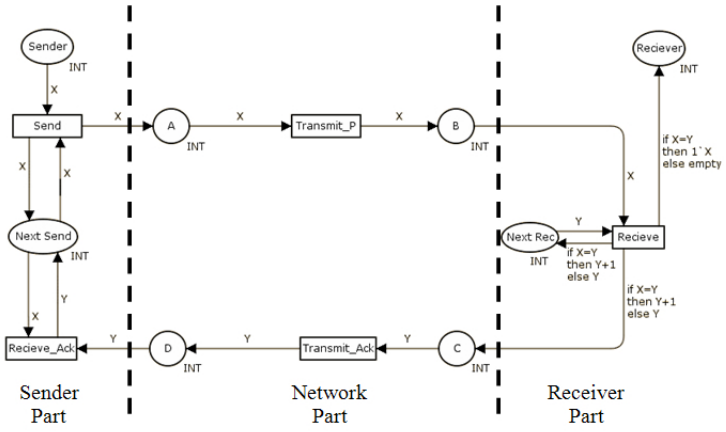



Figure 6. CPN model of the communication protocol

To verify the reachability of the previously given marking ($\langle 1+2+2\rangle.Receiver$), the backward reachability analysis is performed using the inverse CPN, illustrated in Fig.7. Inversions of $Transmit\_P$ and $Transmit\_Ack$ are basic transformations. Those of $Receive\_Ack$ is a case of parametric transformation (The input variable $X$ is not used in an output function). The goal of this transition is to delete the old value in $Next\_Send$ and to replace it by the token value of the place $D$. So, in the inverse reasoning, the goal is to put in $D$ the token value of $Next\_Send$ and to put in $Next\_Send$ an arbitrary value. The token already contained in this place is then returned. The inversion of transition $Send$ is a parallel one. This is can be seen through the input variable $X$ which is used in two output functions (toward $A$ and $Next\_Send$). This explains the new variable $Z$ and the guard $[X = Z]$. The inversion of $Receive$ is a combination of parallel and parametric (the function are parameterized by $X$ and $Y$). In the inverse problem, tokens in $Receive$ are known which means that possible initial values of $X$ are known.

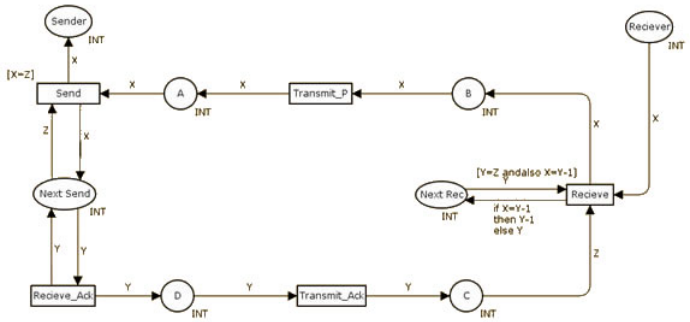



Figure 7. inverse CPN of the communication protocol

In the original CPN, the sequence of transition firing is : $Send$, $Transmit\_P$, $Receive$, $Transmit\_Ack$, $Receive\_Ack$. This corresponds to a partial order of transition firings. The inverse CPN analysis (endowed by the marking $\langle 1+2+2\rangle.Receiver$) must respect the inversed order starting with transition $Receive$ which is the unique potentially valid transition. The $receive$ firing is done using marking enhancement. Two tokens have to be added to the model : the first one into place $Next\_Rec$ and the second into place $C$. The two tokens are numbered 3 because this value is the only one that fulfills guard constraints. This marking enhancement corresponds to the assumption that the packet numbered 2 was correctly received. The result is a token of value 2 in places $Next\_Rec$ and $B$ after firing $Transmit\_P$ in $A$. The firing of $Send$ requiers a token valued 2 in $Next\_Send$ which assumes a correct packet transmission. The marking evolution (without any enhancement) processes the packet numbered 1 thanks to all transitions firings (one time for each one). This leads to a deadlock situation with a packet numbered 2 in place $Receiver$.

Thus the conclusion is reached and expresses that there is no coherent initial marking possible to obtain the $\langle 1+2+2\rangle.Receiver$ marking. Upon detection of such marking, an unexpected fault has to be declared.

## 7. Conclusion

This paper presents an extension of the backward reachability to some categories of CPN. As the backward reachability was not originally defined on the CPN the corresponding rules for are also introduced. The rules of extension, such as

the definition of the colored incidence applications for the inverse CPN, allow to keep the properties and the advantages of the backward reachability and apply them to the CPN class.

The main application field for the backward reachability is the model diagnostics during the conception. The system initial state is usually well determined and so is the final state which in this case represents an undesired state. The proposed method can provide answers to the possibility of undesired event occurrence, the earliest failure time, the system evolution vector, etc.

The perspectives start with the development of an automated tool for a verification of the CPN structure in order to be acceptable for the backward reachability analysis. The theory advances include the generalization of transformation rules to non elementary loops and to composed functions. The second path to explore is the possible of presence of non deterministic time constraints for the transition firing. The study of stochastic firing vectors is also considered as a potential subject of interest.

## References


[1] S. M. Cho, H. S. Hing, and S. D. Cha. Safety analysis using colored petri nets. In *Proc. Asia-Pacific Software Engineering Conference (APSEC-96), 4-7 December 1996, Seoul, Korea*, pages 176–183, 1996.

[2] S. Haddad. A reduction theory for coloured nets. *Lecture notes in Computer science n 424 Springer-Verlag*, pages 209–235, 1989.

[3] K. Jensen and G. Rozenberg. *High-level Petri Nets : Theory and Application*. Springer-Verlag, Germany, 1991.

[4] S. Khalfaoui. *Mthode de recherche des scnarios redouts pour l'valuation de la sret de fonctionnement des systmes mcatroniques du monde automobile*. PhD thesis, Institut National polytechnique de Toulouse (France), Septembre 2003.

[5] Sous la direction M. Diaz. *Les rseaux de Petri : modles fondamentaux*. Herms Science Publications, Paris (France), 2001.

[6] N. G. Leveson and J. L. Stolzy. Safety analysis using petri nets. *IEEE Trans. Softw. Eng.*, 13(3):386–397, 1987.

[7] N. Metropolis and S. Ulam. The monte carlo method. *Journal of American Statistical Association*, 44(247):335–341, September 1949.

[8] Carl Adam Petri. *Communication with automata*. PhD thesis, Darmstadt Institut fr Instrumentelle Mathematik, Bonn (Germany), 1962.