University of La Rioja

Doctoral Thesis

Security in Petri nets sharing and storage: subnets, privacy, integrity, authentication and non repudiation

Author:
Iñigo León

Supervisor:
Dr. Emilio Jimenez

A thesis submitted in fulfilment of the requirements
for the degree of Computer Science

in the

Research Group Name
Department of Electrical Engineering

June 2015

Declaration of Authorship

I, Iñigo León, declare that this thesis titled, 'Security in Petri nets sharing and storage: subnets, privacy, integrity, authentication and non repudiation' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: June 2015 _____

"Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism."

Dave Barry

UNIVERSITY OF LA RIOJA

Abstract

Faculty of Science, Agrifood Studies and Computing
Department of Electrical Engineering

Doctor of Philosophy

Security in Petri nets sharing and storage: subnets, privacy, integrity, authentication and non repudiation

by Iñigo León

In this thesis I approach the study of Petri nets from the point of view of the security. There several goals in this thesis. First of all, I will define a subnetting process by building a framework of definitions and notations to create subnets from the original Petri net. Then, the creation of a PNML extension that allows the representation of subnets. In this work only the structure of the network is processed. The study of markings and properties of nets with hidden pieces will we analyzed in further works.

One application of this subnetting and PNML representation is the possibility of hiding part of a Petri net, facing a possible distribution, maintaining the privacy of the critical, secret, or complex parts of the system. However this information is not eliminated from the net, but encrypted inside. Other application explained is the possibility of digital signature of subnets, providing security services to the net and/or subnets.

My original contribution to knowledge are:

1. Comprehensive study of subnets, defining interfaces that abstract their internal structure from the exterior by using interfaces. A method to build these subnets for the complete Petri net is explained and analyzed matrixed.

2. PNML has no way to represent subnets, so I approach a possible PNML extension to do it.

3. Applications of subnetting and PNML extension to represents subnets allow me to apply several security technics that offers encryption, data integrity, authentication and non repudiation

Acknowledgements

//BORRAR The acknowledgements and the people to thank go here, don't forget to include your project advisor...

Thanks to my years of personal study and my curiosity for new knowledge, I fought to combine my job, my family and studies. I has been a difficult work, but the result has been worth it.

//TODO

Contents

List of Figures

Abbreviations

PN	**P**etri **N**et
SN	**S**ub**N**et
N	**N**et
PIM	**P**laces **I**nfluence **M**atrix
TIM	**T**ransitions **I**nfluence **M**atrix
...	//TODO

For/Dedicated to/To my...

Chapter 1

Introduction

//TODO Poner tambien las keywords

1.1 Background of the research

Petri nets are widespread for modeling many classes of systems, such as manufacturing logistics processes and services [1, 2], concurrent systems [3]. However, all these nets are described in a comprehensive way and must have the information of the entire net to determine its evolution.

1.2 Research problem

The problem occurs when somebody doesn't want to describe the whole subnet. Or, maybe, is wanted that one part of the process is only accessible for one specific person or entity.

The first approach to solve this problem is to take two Petri nets

- One Petri Net with the public information, extracting the private data. This is an incomplete model of the process
- Another Petri Net with the whole information for the interested person or entity.

As you can notice, this is not an efficient way to publish this kind of Petri Nets.

1.3 Justification of the research

It would be interesting to provide security to a Petri net:

- hiding a part of it. This can be useful, for example, distributing a process we want to be secret [4], or simply to be a part of the net to be complex and do not interested handle for any reason [4].
- avoiding not allowed changes in it (or a part of it).
- authenticating it (or a part of it). Useful to ensure who has developed a Petri net or subnet.
- avoiding the possibility of supplant other people in the authority of the Petri net or some of its parts.

So here is my contribution. I have researched the possibilities of hiding a part of a Petri Net so that everybody can access the public information, maintaining the secret of the private data. This private data is accessible only for authorized people. And not only that: I ensure data integrity, authentication and non repudiation to Petri nets or subnets.

Some authors study the possibilities of Petri nets reduction [5, 6, 7, 8, 9, 10], grouping in one place or transition a subnet, so that what happens on this subnet, is encapsulated in a single point of execution. However, we want to go further by defining parts of the net that are hidden (not clustered) and what are the implications, studied within network properties.

The main objective of this thesis is to take parts of a Petri nets and provide them of wide security (privacy, integrity, authentication and non repudiation).

1.4 Methodology

In order to achieve this goal, I have defined three milestones:

1. Extend Petri Nets definition in order to define subnets, abstracting the internal structure from the the rest of the net, focussing on hiding information.
2. Choose a lossless and extendible representation of this kind of Petri Nets
3. Define a hiding and signing method for this representation

For the first milestone, I work for the creation of the theoretical basis for further study of Petri nets in which certain parts are hidden. So we setup a generic framework of definitions and notations that allow us to deepen in the study of the characteristics and properties of Petri nets and their subnets [11, 12]. Also mention work already carried by other researchers in which we rely for our goal (i.e. [3, 13, 14, 15]). All this will be necessary to create the framework that allows us to study occultation in Petri nets. We will expand the vision of Petri nets, providing them with greater functionality, such us attachable subnet.

The next step in this work is to choose (or define) a flexible representation of Petri nets that allows us to translate the previous extended nets. This representation has to be really extendible and flexible in order to be able to show actual and future characteristics of Petri nets. I can advance you that the selected representation is the standard PNML and I have to define and extension for it in order to represent subnets that are going to be hidden.

Once selected this representation, the last step is the hiding and signing method. Once more, I bet for standard protocols like XMLEncription and XMLSignature.

This is a very basic investigation because I extend the very early definitions of Petri nets. Because of it, the results of this thesis is very probably extensible to any other development whose base are the classic Petri Nets. For example, I am not going to study colored Petri nets, neither timed Petri nets, etc. But it is very easy to see that the results achieved in this thesis can be applied to them with no problem.

1.5 Delimitations of scope and key assumptions

For this work we will always deal with ordinary and pure networks, unless otherwise expressly. This assumption is only for clarity reasons, because the protocols and methods described in this work are perfectly extensible to other kind of Petri nets, as long as these Petri nets are representable in PNML format.

Chapter 2

Literature review

In this chapter I am going to go over the ancient Petri net history. This work is a very basic investigation on Petri nets. This means that the most of the references are quite general.

For this literature review, I will follow the structure of this thesis:

1. First of all, I am going to describe some generalities of Petri nets as an introduction
2. Then I will describe subnets and the process of splitting Petri nets into several subnets. Some of those subnets (or the complete net) are going to be secured.
3. The next step is to explain some possible Petri net representations and my selection of PNML for the securizing process
4. Once selected PNML, I am going to extend this language to support the subnets defined before
5. The last step is the securing properly speaking. To do this, I will use the standard XMLEncryption for ciphering the secret information and XMLSignature for integrity, authentication and non repudiation.

2.1 Introduction. Petri nets

In the 60's, Carl Adam Petri invented a new way to describe distributed systems called Petri nets [16, 17, 18]. Many net theories are based on those works[19]. Nowadays, Petri nets are really extended to represent discrete systems [20, 21, 22, 23, 24]. There are lots of applications of Petri nets:

- Modelling of sequential processes[25], concurrent systems[26, 27], manufacturing [15, 28, 29, 30, 31], logistic processes [2], discrete event systems [32], ...
- Simulation of industrial applications [33, 34], logistic and production systems [1],...

This work is not about of Petri net application, so I am not going to deepen this field. However I really mind the intrinsic structure of them. Since the definition of Petri nets, many authors investigated about them. The basic basis of my work are some of the best Petri net researchers, who are included in this review:

- Murata, T [11, 35, 36]
- Silva, M [12, 31, 37]
- Peterson, JL [14]

And not only general Petri nets. Any kind of extensions are well received

- Lien [38]: Generalized Petri nets
- Jensen, K [3, 26, 39]: High level Petri nets, coloured Petri nets
- Silva [23, 40]: Continuous Petri nets
- Khomenko, V [41]: High level Petri nets
- Ratzer [42]: Coloured Petri nets
- Kristensen [27, 43]: Coloured Petri nets
- Silva [44]: Fluid Petri nets
- Latorre [20, 22]: Aggregation Petri nets and coloured Petri nets
- Campos [45]: Stochastic Petri net models
- Recalde [46]: Continuous Petri nets
- David [13]: Discrete, continuous and hybrid Petri nets
- Fraca [47]: Fluid and untimed Petri nets
- Vazquez [48, 49]: Stochastic continuous Petri nets and fluid Petri nets

The study of subnetting and securing of all of them are outside the scope of this work, but as all of these Petri nets extensions are reproducible by a graph, they are susceptible of applying the same described methods with light variations (like color, arc types, ...).

Several authors studied properties of Petri nets and they have been very useful as [50, 51, 52, 53, 54]. But they don't contribute in the main goal of my work. They are general references for the theoretical development.

Other fonts are authors that studied general theory on structure of systems, as Teruel [55] is.

But this work is not oriented towards studying all of these Petri net type. I want my work so general that it is easily exported to practically any kind of existing Petri nets or future Petri nets types that are not defined still.

2.2 Subnets

The study of subnets is very ancient with general works by Silva [12, 31], Murata [11, 36], Peterson [14]. All of them have been very useful for my preparation and they are the theoretical basics of my work. But there are other more specific works that study several aspects of subnets.

The first approach was presented by Valette [9], who studied the possibility and properties of replacing of places or transitions by subnets. Suzuki (in collaboration with Murata) [8] continued it with a method for expanding and reducing Petri nets. Basically, they wanted to substitute places and transitions by subnet and viceversa, maintaining the properties of the net. Druzhinin (and Yuditskii) [5] completed these works explaining how to construct regular Petri nets from standard subnets. But basically is an algorithm to replace places or transitions with well-formed Petri nets. My contribution in this field is the description of a way to replace subnets (not only places or transitions) with other subnets. Their work can be seen as a particular case of my description of macroplace an macrotrasition. However I don't deepen the properties because it is not my goal.

Other works important in this section are Fahmy's ones [6, 7]. At first sight (only reading the title), it seems to be the same I explain in the chapter 3. But this is not true. These two articles by Famhy are about the analysis of large Petri nets by cutting them into pieces. The second one [7] is and extension of the first one [6]. The idea is "divide and conquer" over the net. The partitioning of a net is useful because it preserves the properties of the original whole net. He centers the interest on the characteristics of the net and the partitions. But he doesn't make an extensive study of subnets, that is what

I do in my work. Basically, it is something like the previous paragraph: it can be seen as a particular case of my study.

Other work of interest is Hsieh' one [56], where he analyzes non-ordinary Petri nets for flexible assembly/disassembly processes based on structural decomposition, but it is not a net decomposition but a process decomposition. So it has no more interest for my work.

The last entrance in this section is the work of Xia [10]. His objective is to encapsulate a subnet in one place or transition. Then he study several ways of simplify that subnet or erasing redundant information. These subnets are grouped into a single point of execution. I want to go further by defining hidden (not grouped) parts of the net. However, this work has much to do with macroplaces and macrotransitions introduced in chapter 3.

2.3 Petri net representation

Other important question in this thesis is the election of a Petri net representation that allows us to translate defined subnets into that format in order to apply posterior operations over those subnets.

Apart from the general works of Petri nets named before introduced by Petri, Murata, Peterson and Silva [11, 12, 14, 16, 17, 18, 31, 35, 36, 37], there are other specific articles in this field:

- Hura [57] introduced the state space representation of Petri nets, but with no possibility of subnet representation
- Anishimov and Perchuk [58] in their work "Representation of exchange protocols and Petri using finite sequential machine nets"didn't do exactly what the title says. They used High level Petri nets in order to define interaction protocols between two objects. These protocols are seen as more precise formal languages. Definitely, nothing seemed to my goal.
- Das [59] defined the reflexive incidence matrix (RIM) representation of Petri nets, The possibilities of represent subnets in this case is basically the same as normal incidence matrices. So my method can be applied in the same way with this representation.
- Malyugin [60] worked in an arithmetical representation of Petri nets, but, as I said with Hura, there is no easy implementation of subnet representation. So this representation is not useful for my work.

- Kaushal [61] introduced a new formulation for state equation representation for Petri nets. The problem is the same as with Hura and Malyugin: no subnet representation.
- Kiritsis and Xirouchakis [62] defined a new matrix implementation of Petri nets for process planning. As it is matrix representation my method can be applied in the same way, as I said with Das.

2.4 PNML

PNML is a standard for representation of Petri nets. Because of that, the main source of information is internet. In particular the official site `www.pnml.org` [63] has all the necessary information to work with it.

After several years of study, in 2004 the ISO/IEC 15909-1 [64] appeared to define conceptually and mathematically a xml representation of Petri nets: PNML [65]. It is explained in a less formal way by Billington [66].

So all the Petri nets that I am able to draw can be stored in PNML. Because of that, it is one of the most supported formats in almost every program that draw Petri nets.

PNML [63] is an implementation defined by the standard ISO/IEC 15909-2:2014 [67] that complement the anterior ISO/IEC 15909-1. The goal of this standard ISO is to define a transfer format of Place/Transition nets, High-level Petri nets and Symmetric nets. However, it is designed in such way that it can be easily extended, so that other versions of Petri nets can be supported later.

2.5 PNML extensions

There are ways to define these extensions in a graphical way, using eclipse as intermediate named ePNK, explained by Kindler [68] and Hillah [69]. Additionally, Moutinho [70] and Ribeiro [71] defined several PNML extensions for several kinds of Petri nets.

But this is not the main goal of my work. My intention is to describe the subnet extension in a theoretical way. Once described, it could be implemented with that kind of tools.

2.6 Securize information on Petri nets

There is very little literature about this topic (except my own work [4]). There are a couple of articles that seems to threat the theme, but with a deeper study we can see that the achievements are not what I am looking for.

Mahulea [72] point the problem of observability of timed CPNs, but not with the same meaning of my vision. His problem is that in determined circumstances it is difficult to estimate the initial/ actual state/marking of a Petri net. Anything like my work.

By his side, Saabori [73] approach the topic of hiding but not in Petri net structure, but in states of the Petri net. So it doesn't help.

Other work that at first sight may be interesting is one article from Velilla [74] that define a mechanism for safe implementation of concurrent systems. But this safety is for the process itself, not for the safety of the implementation. So it doesn't contribute to my work.

My intention is to securize part of a Petri net in two ways: hide and sign. The way to achieve this goal is cutting the net in subnets, represent them in PNML and then one or more of this goals:

- hide it
- avoid unwanted changes
- ensure identity of the custodian of the Petri net
- avoid the impersonation of the custodian

But there is not literature about the PNML representation for subnets, so this is new knowledge.

2.7 Security: XMLEncryption and XMLSignature

Both XMLEncryption and XMLSignature are W3C recommendations.

XMLEncryption it is a way to cipher xml content. And not only that. Actually, it is a way to cipher any kind of information and store this ciphered content in xml format. In the same way, XMLSignature is a standard mode to sign xml or non xml content.

There are thousands of articles and applications of it, but, as they are W3C recommendation, they are completely defined. The last version of XMLEncryption and XMLSignature definition are [75, 76] respectively.

In this work, I apply these technologies in order to achieve the next goals

- Privacy: hide a entire or a part of a petri net
- Integrity: nobody should modify concrete parts of a Petri net without being detected.
- Authentication: guarantee that nobody can impersonate another, for example stealing the creation of a Petri net or part of it.
- Non repudiation: ensure that nobody but a concrete person has done something in a Petri net (or part of it), for example, sign a Petri net.

In particular, encryption provides privacy. Digital signature provides integrity, authentication and non repudiation.

My contribution to knowledge in this case is to mix two different knowledge areas as Petri nets and information security (encryption/digital signature) are.

So there are no literature about this topic, because nobody has developed it until I do.

Chapter 3

Private information in Petri nets. Subnets

3.1 Introduction

By default, a Petri net is described in a comprehensive and public way. In this chapter I am going to describe a way to declare private information of a Petri net. This information is candidate to be hidden, complying in this way with the goal of privacy. Furthermore, this theory can be applied afterwards in order to sign parts of a Petri net (subnets) achieving the objectives of integrity, authentication and non repudiation.

First of all I have to create a like framework of definitions, properties and methodologies in order to achieve this goal. The main idea is the concept of subnet an its interaction front-end.

Once defined this subnets, the Petri net can be divided into public and private chunks only by ordering the places and transitions in one subnet or another.

3.2 Petri subnets

3.2.1 Definitions and properties

Let P and T the non-empty finite sets of places and transitions of a Petri net, respectively. Let $|P| = n$ (the number of places of the net) and $|T| = m$ (number of transitions of the net). Let α and β pre and post incidence matrices respectively. Let $N = \langle P, T, \alpha, \beta \rangle$ be a Petri net and let C the incidence matrix of N

Definition 3.1 (Subnet [12])**.** A subnet of $N = \langle P, T, \alpha, \beta \rangle$ is a net $\overline{N} = \langle \overline{P}, \overline{T}, \overline{\alpha}, \overline{\beta} \rangle$ so that $\overline{P} \subseteq P$ y $\overline{T} \subseteq T$, $\overline{\alpha}$ y $\overline{\beta}$ are restrictions of α and β over $\overline{P} \times \overline{T}$.

In other words, a subnet is a subset of places and transitions together with the arches that connect them together.

Let's look at the implications of the latter definition since it is one of the most important with regard to this work.

A subnet corresponds [4], in terms of matrices, with the resulting submatrix keeping only the rows and columns corresponding to places and transitions of the selected subnet.

Example 3.1. *Let's take the Petri net which has the following incidence matrix:*

$$C = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{array} \begin{array}{c} \begin{array}{cccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 \end{array} \\ \begin{pmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} \end{array}$$

If we select the places p_1, p_3, p_4 and p_5 and transitions t_1, t_2, and t_6 we have the subnet defined by this incidence matrix

$$C' = \begin{array}{c} \\ p_1 \\ p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{ccc} t_1 & t_2 & t_6 \end{array} \\ \begin{pmatrix} -1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \end{array}$$

by simply erasing p_2 and p_6 rows and t_3, t_4 and t_5 columns.

In [4] is shown that the set of all possible permutations of rows and/or columns of a matrix of incidence corresponding to a Petri net, either the pre or post incidence matrix, make an equivalence relation. In other words, given an incidence matrix, both rows and columns can be rearranged and this rearrangement describes exactly the original Petri net.

In this way, we can study the incidence matrices reordering rows and columns as preferred one at any time, without loss of generality.

$$
\begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \end{array}
\begin{array}{c}
\begin{array}{cccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 \end{array} \\
\left(\begin{array}{cccccc}
-1 & 0 & 1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & -1 & 1 \\
0 & 0 & 0 & 0 & 1 & -1
\end{array} \right)
\end{array}
\equiv
\begin{array}{c} \\ p_8 \\ p_1 \\ p_3 \\ p_6 \\ p_4 \\ p_5 \\ p_2 \\ p_7 \end{array}
\begin{array}{c}
\begin{array}{cccccc} t_1 & t_6 & t_3 & t_5 & t_4 & t_2 \end{array} \\
\left(\begin{array}{cccccc}
0 & -1 & 0 & 1 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 1 \\
-1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & -1 & -1 & 0
\end{array} \right)
\end{array}
$$

FIGURE 3.1: Two equivalent incidence matrices to describe the same Petri net

From all these definitions and proofs we can draw several trivial conclusions:

1. A subnet, like generic Petri net does not have to be square.
2. If a row or column of the incidence matrix of the subnet is all zeros, it doesn't mean that that place or that transition is isolated. This occurs only with pure nets. If there is an arc from one transition to a place and an arc from that place to the same transition with the same weight, the sum (the matrix element associated to that place/transition) is zero , but there are two arcs.
3. It does not matter the number of places and/or transitions that are chosen for the subnet, as long as they are not empty sets.

3.2.2 Splitting a net into subnets

Let $N = \langle P, T, \alpha\beta \rangle$ a Petri net where $|P| = n$ and $|T| = m$. So $P = \{p_1, p_2 ... p_n\}$ and $T = \{t_1, t_2 ... t_m\}$. Select two subsets $P' \subseteq P$ and $T' \subseteq T$ so that $|P'| = r \leq n$ and $|T'| = s \leq m$. With these premises divide into two subnets the original one.

We have seen that we can identify a subnet by simply removing rows and/or columns (places/transitions) of an incidence matrix. Taking advantage of the equivalence relation defined in [4], we reorder the incidence matrix so that places and transitions of the subnet are in the top-left. Without loss of generality, and for convenience, places and transitions

can be renamed so that the incidence matrix is as follows:

$$C = \begin{array}{c} \\ p_1 \\ \vdots \\ p_r \\ p_{r+1} \\ \vdots \\ p_n \end{array} \begin{array}{c} \begin{array}{ccc|ccc} t_1 & \cdots & t_s & t_{s+1} & \cdots & t_m \end{array} \\ \left(\begin{array}{ccc|ccc} a_{11} & \cdots & a_{1s} & a_{1(s+1)} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{r1} & \cdots & a_{rs} & a_{r(s+1)} & \cdots & a_{rm} \\ \hline a_{(r+1)1} & \cdots & a_{(r+1)s} & a_{(r+1)(s+1)} & \cdots & a_{(r+1)m} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{ns} & a_{n(s+1)} & \cdots & a_{nr} \end{array}\right) \end{array}$$

We now have the net divided into two disjoint and complementary subnets. They are disjoint because there is no place and no common transition, and complementary because the union of the two we gives the complete net. At this point note that the incidence matrix is divided into four blocks $C = \begin{pmatrix} N_1 & PIM_{12} \\ TIM_{12} & N_2 \end{pmatrix}$. The interpretation is as follows:

- N_1 subnet made up of places $p_1..p_r$ and transitions $t_1..t_s$
- N_2 subnet that is complementary to N_1, made up of the places $p_{r+1}..p_n$ and transitions $t_{s+1}..t_m$
- PIM_{12} (Places Influence Matrix) is the matrix that defines the interaction of the N_1 places with N_2 transitions. Basically it is the matrix whose elements are those that are in the same rows of N_1 but outside of it (rows $1..s$ and columns $s+1..m$).
- TIM_{12} (Transitions Influence Matrix) is the matrix that defines the interaction of N_1 transitions with N_2 places. It is the matrix whose elements are in the same columns of N_1 elements but outside of it (rows $r+1..n$ and columns $1..s$).

We can notice that $PIM_{12} = TIM_{21}$ and $PIM_{21} = TIM_{12}$ by applying the definition.

This can be generalized to multiple disjoint and complementary subnets without further to re-apply the same process to any of the subnets already defined. Thus, generically

we can divide a network into i subnetworks, so we'll have a matrix of this style:

$$\left(\begin{array}{ccc|ccc|c|ccc}
a_{11} & \cdots & a_{1s} & a_{1(s+1)} & \cdots & a_{1t} & & a_{1u} & \cdots & a_{1m} \\
\vdots & \mathbf{N_1} & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\
a_{p1} & \cdots & a_{ps} & a_{p(s+1)} & \cdots & a_{pt} & & a_{pu} & \cdots & a_{pm} \\
\hline
a_{(p+1)1} & \cdots & a_{(p+1)s} & a_{(p+1)(s+1)} & \cdots & a_{(p+1)t} & & a_{(p+1)u} & \cdots & a_{(p+1)m} \\
\vdots & \ddots & \vdots & \vdots & \mathbf{N_2} & \vdots & \cdots & \vdots & \ddots & \vdots \\
a_{q1} & \cdots & a_{qs} & a_{q(s+1)} & \cdots & a_{qt} & & a_{qu} & \cdots & a_{qm} \\
\hline
 & \vdots & & & \vdots & & \ddots & & \vdots & \\
\hline
a_{r1} & \cdots & a_{rs} & a_{r(s+1)} & \cdots & a_{rt} & & a_{ru} & \cdots & a_{rm} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \mathbf{N_i} & \vdots \\
a_{n1} & \cdots & a_{ns} & a_{n(s+1)} & \cdots & a_{nt} & & a_{nu} & \cdots & a_{nm}
\end{array}\right)$$

In this situation, if we select two subnets N_j and N_k, we locate the zones of influence of each with respect to the other:

$$\left(\begin{array}{c|c|c|c|c}
\ddots & \cdots & \cdots & \cdots & \cdots \\
\hline
\vdots & SN_j & \cdots & PIM_{jk} = TIM_{jk} & \cdots \\
\hline
\vdots & \cdots & \ddots & \cdots & \cdots \\
\hline
\vdots & TIM_{jk} = PIM_{kj} & \cdots & SN_k & \cdots \\
\hline
\vdots & \vdots & \cdots & \vdots & \ddots
\end{array}\right)$$

Thus, the submatrix PIM_{jk} represents the arcs that connect places of the submatrix N_j with N_k transitions and the matrix TIM_{kj} represents the arcs that connect places of SN_k to SN_j transitions.

Notation. For simplicity, we will call

- PIM_i to all the elements in the same rows of N_i but outside of it.
- TIM_i to all the elements in the same columns of N_i but outside of it.

We can notice again that $PIM_{jk} = TIM_{kj}$ and $PIM_{kj} = TIM_{jk}$ by applying the definition.

Definition 3.2 (Partition of a Petri net into subnets). We say that a set $P = \{N_1 N_2 ... N_k\}$ is a partition into subnets of N if the following holds:

- $N_1 \cup N_2 \cup ... \cup N_k = N$
- $\forall i, j | 1 \leqslant i, j \leqslant k \Rightarrow \ N_i \cap N_j = \emptyset$ (pairwise disjoint)

3.2.3 Subnet classification

Depending on how each of the four pieces of matrix (N_1, N_2, PIM and TIM) are, we can see some special cases.

3.2.3.1 Disjointed subnets

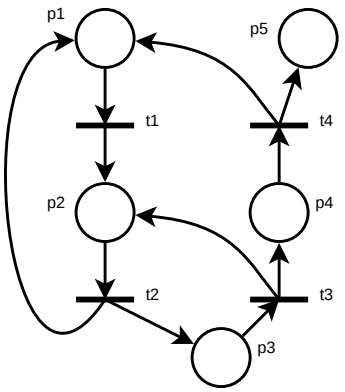
Definition 3.3 (Disjointed subnet). Pure subnet said disjointed if there is no arc between its own places and transitions

Suppose that in the incidence matrix divided into the four pieces explained, are N_1 or N_2 be a null matrix. In this case the interpretation is that there arcs between places and transitions of the subnet, which would simply places and/or no transitions related to each other but with the additional subnetwork. Subnet talk then disjointed.

Proposition 3.4. *A subnet is disjoint iif its incidence submatrix is the null matrix.*

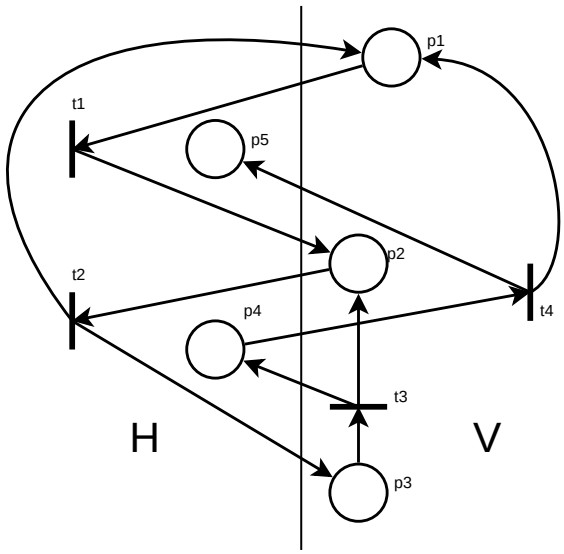
The demonstration is trivial.

Example 3.2. *Consider the next well-printed Petri net and its incidence matrix:*



$$C = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{pmatrix} -1 & 1 & 0 & 1 \\ 1 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

We assume that we select as subnet the one formed by 4th and 5th places and transitions 1 and 2. Then the graph and the incidence matrix are thus:



$$C = \begin{array}{c} \\ p_4 \\ p_5 \\ \hline p_1 \\ p_2 \\ p_3 \end{array} \begin{array}{c} \begin{array}{cc|cc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \left(\begin{array}{cc|cc} 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \\ \hline -1 & 1 & 0 & 1 \\ 1 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{array}\right) \end{array}$$

Here we can see that although really p_4, p_5, t_1 and t_2 are not isolated, there is no arc that connects them together. In the incidence matrix, the corresponding submatrix is the zero matrix. Therefore, whether or not there are elements isolated in the net, total subnet formed by p_4, p_5, t_1 and t_2 is a disjointed subnet.

We can extend this definition to a subnet that is part of a bigger net. It doesn't matter in how many subnets it is separated: if one subnet is the null matrix, this subnet is disjointed. This characteristic is implicit to the selected subnet and it doesn't depend on the rest of the net.

3.2.3.2 Macroplace

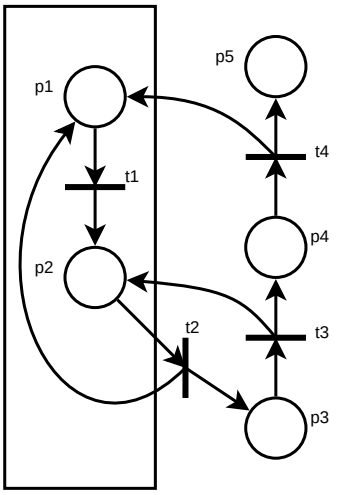
Once subnet concept is introduced, we can classify them by some properties. In this case, a subnet which behaviour is like a place, relating only to transitions can be defined as macroplace.

Definition 3.5 (Macroplace). A macroplace is a subnet that meets the following:

1. arcs entering any node of the subnet from an external node come from a transition.
2. arcs leaving any node on the subnet to an external node go to a transition.

Proposition 3.6. *A subnet N_1 is a macroplace iif its transitions influence matrix (TIM) is the null matrix. In the same way, N_2 is a macroplace iif its places influence matrix (PIM) is the null matrix*

$$N_1 \textit{ is macroplace} \iff \forall a_{ij} \in TIM_1, a_{ij} = 0$$



$$C = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \left(\begin{array}{cccc} -1 & 1 & 0 & 1 \\ 1 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{array}\right) \end{array}$$

FIGURE 3.2: Macroplace

$$N_2 \text{ is macroplace} \iff \forall a_{ij} \in PIM_1, a_{ij} = 0$$

The demonstration is trivial.

Suppose that the incidence matrix divided into the four pieces explained, TIM appears to be the zero matrix. Then we conclude that the subnet N_1 is only related by arcs with places of subnet N_2. All arcs entering N_1 come from transitions of N_2 and all arcs coming out from N_1 go to transitions of N_2. Stated another way, the subnet N_1 behaves like a place, but may contain places and transitions.

Note that this is not really a place, and that the subnet has not marked as such. The marking is on the places within the subnet and depends on the arches of arrival.

Example 3.3 (Macroplace). *If we take the figure 3.2 we can see that the subnet inside the rectangle N_1 is related to the rest of the net only through transitions, it is to say, there is no arc from an external place or from an internal transition. Furthermore, TIM_1 is the null matrix. So this subnet is a macroplace.*

We can extend this definition to a subnet that is part of a bigger net too. However, in this time, it does matter the way in which the bigger net is separated. This characteristic is not implicit to the selected subnet and depends on the rest of the net.

//TODO Estos tres siguiente TODO no son estrictamente necesarios. Hacerlo solo si da tiempo

//TODO explicar macroplace dependiendo del resto de subredes. Una subred puede comportarse como macrolugar sobre otra subred, pero no sobre una tercera.

//TODO macroplace absoluta si da igual el resto de subredes

//TODO Caracterizacion de macroplace absoluta

3.2.3.3 Macrotransition

In the same way as macroplaces, a subnet which behaviour is like a transition, relating only to places can be defined as macrotransition.

Definition 3.7 (Macrotransition). A macrotransition is a subnet that meets with the following:

1. arcs entering any node of the subnet from an external node come from a place.
2. arcs leaving any node on the subnet to an external node go to a place.

Proposition 3.8. *A subnet N_1 is a macrotransition iif its places influence matrix (PIM) is the null matrix. In the same way, N_2 is a macrotransition iif its transitions influence matrix (TIM) is the null matrix*

$$N_1 \text{ is macrotransition} \iff \forall a_{ij} \in PIM_1, a_{ij} = 0$$

$$N_2 \text{ is macrotransition} \iff \forall a_{ij} \in TIM_1, a_{ij} = 0$$

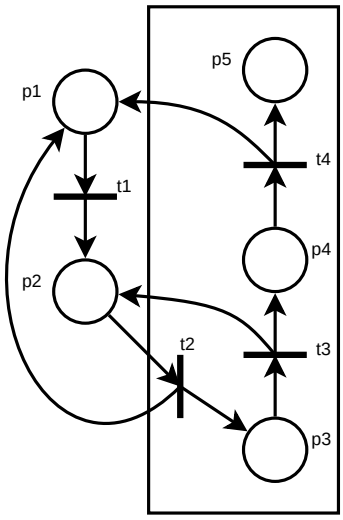
The demonstration is trivial

This is other option that can happen is that in the incidence matrix: PIM appears to be the zero matrix. Then we conclude that the subnet N_1 is only related by arcs with places of subnet N_2. All arcs entering N_1 come from places of N_2 and all arcs coming out from N_1 go to places of N_2. Stated another way, the subnet N_1 behaves like a transition, but may contain places and transitions.

Like macroplaces, macrotransitions are not transitions as such. It is not necessary that all entries are marked to fire the macrotransition, and not all output places are marked after entering it. Everything depends on the inner workings of the macrotransition.

Example 3.4 (Macrotransition). *If we take the figure 3.3 we can see that the subnet inside the rectangle N_2 is related to the rest of the net only through places, it is to say, there is no arc from an external transition or from an internal place. Furthermore, TIM_1 is the null matrix. So this subnet is a macrotransition.*

We can extend this definition again to a subnet that is part of a bigger net too. Like with macroplaces, it does matter the way in which the bigger net is separated.



$$C = \begin{array}{c} \\ p_1 \\ p_2 \\ \hline p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \left(\begin{array}{cccc} -1 & 1 & 0 & 1 \\ 1 & -1 & 1 & 0 \\ \hline 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{array}\right) \end{array}$$

FIGURE 3.3: Macrotransition

//Estos cuatro siguientes TODO no son estrictamente necesarios. Si da tiempo

//TODO explicar macrotransition dependiendo del resto de subredes. Una subred puede comportarse como macrotransition sobre otra subred, pero no sobre una tercera.

//TODO macrotransition absoluta si da igual el resto de subredes

//TODO Caracterizacion de macrotransicion absoluta

//TODO Explicar relacion entre una macrotransicion y un macrolugar en la misma red, con subredes interrelacionadas por esta caracteristica. Una subred es macrolugar sobre otra si y solo si esta ultima es macroplace sobre la primera. Comprobarlo y demostrarlo en su caso

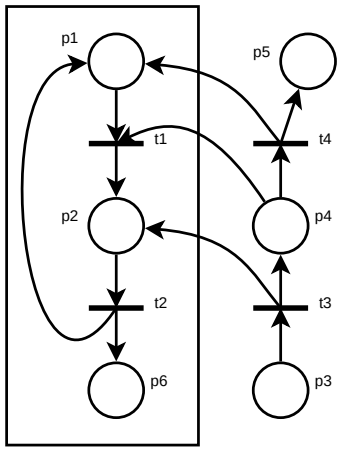
3.2.3.4 Sinkhole subnet

Another thing that can happen is that arrive only arcs to the selected subnet. Then we find that you can not leave the subnet. We speak then of a sinkhole subnet.

Definition 3.9 (Sinkhole subnet). It is said that a subnet is a sinkhole subnet if no arc has its origin in an internal node (place or transition) of the subnet.

It is easy to see that a subnet is sinkhole if and only if all elements of PIM are greater or equal to zero and all elements of TIM are less than or equal to zero.

$$N_1 \text{ is sinkhole} \iff \forall a_{ij} \in PIM_1, a_{ij} \geq 0 \wedge \forall a_{pq} \in TIM_1, a_{pq} \leq 0$$



$$C = \begin{array}{c} \\ p_1 \\ p_2 \\ p_6 \\ \hline p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{cc|cc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \left(\begin{array}{cc|cc} -1 & 1 & 0 & 1 \\ 1 & -1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & -1 & 0 \\ -1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{array}\right) \end{array}$$

FIGURE 3.4: Sinkhole subnet

The demonstration is trivial

Example 3.5 (Sinkhole Subnet). *If we have a look to the figure 3.4, we can see that there is no arc leaving the subnet* N_1 *defined by the rectangle. Furthermore,* $\forall a_{ij} \in PIM_1, a_{ij} \geq 0 \wedge \forall a_{pq} \in TIM_1, a_{pq} \leq 0$*, so we have a sinkhole subnet.*

If we can extend this definition to a subnet that is part of a bigger net, in this case, it does matter the way in which the bigger net is separated. This characteristic is not implicit to the selected subnet and depends on the other subnets.

//TODO reescribir este parrafo anterior para que no sea igual que los anteriores

//TODO estos tres siguiente TODO no son estrictamente necesarios. Hacerlo solo si da tiempo

//TODO explicar sinkhole dependiendo del resto de subredes. Una subred puede comportarse como sinkhole sobre otra subred, pero no sobre una tercera.

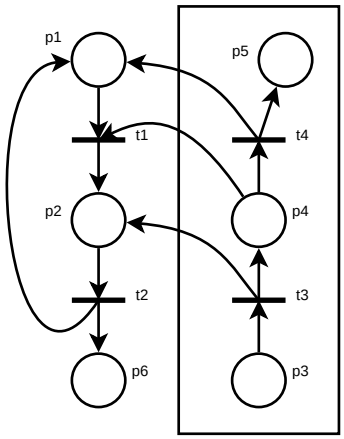
//TODO Sinkhole absoluta si da igual el resto de subredes

//TODO Caracterizacion de Sinkhole subnet absoluta

3.2.3.5 Source subnet

If instead of this what happens is no arc gets into the subnet, we have a source subnet. In a source subnet we can not enter.

Definition 3.10 (Source subnet). It is said that a subnet is a source subnet if no arc has its destination in an internal node (place or transition) of the subnet.



$$C = \begin{array}{c} \\ p_3 \\ p_4 \\ p_5 \\ \hline p_1 \\ p_2 \\ p_6 \end{array} \begin{array}{c} \begin{array}{cc|cc} t_3 & t_4 & t_1 & t_2 \end{array} \\ \left(\begin{array}{cc|cc} -1 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 1 & -1 & 1 \\ 1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{array}\right) \end{array}$$

FIGURE 3.5: Source subnet

It is easy to see that a subnet is source if and only if all elements of TIM are greater than or equal to zero and all elements of PIM are less than or equal to zero.

$$N_1 \text{ is source} \iff \forall a_{ij} \in TIM_1, a_{ij} \geq 0 \wedge \forall a_{pq} \in PIM_1, a_{pq} \leq 0$$

The demonstration is trivial

Example 3.6 (Source Subnet). *In the same way that with the sinkhole subnet explained before, if we take the figure 3.5 we can see that the subnet N_1 inside the rectangle has no arc leaving it and $\forall a_{ij} \in TIM_1, a_{ij} \geq 0 \wedge \forall a_{pq} \in PIM_1, a_{pq} \leq 0$ so we have a source subnet.*

3.2.4 Matrix parts description once defined the subnets

As places and transitions can be reordered smoothly, we study a network N divided into 2 subnets, for simplicity and without loss of generality. At this moment I am going to study several topic in order to approach privacity, by hiding a subnet. For consistency with [4] we will follow this notation:

$$\left(\begin{array}{c|c} H & HP \\ \hline HT & V \end{array}\right)$$

where

- H (Hidden Subnet) is the subnet wanted to be hidden.
- V (Visible Subnet) is the subnet that is visible.

- HT (Hidden Transitions Submatrix) are the relationships between places of V and H transitions
- HP (Hidden Places Submatrix) are the relations between transitions of V and H sites

Note. Following this notation can be convenient because it is clear what is each of the submatrices. However, elsewhere in the document be referenced as N_1 and N_2 for be more clarifying or being something generic and independent networks concealment. However, using N_1 and N_2 the notation of subnets of influence with respect to the other is more diffuse.

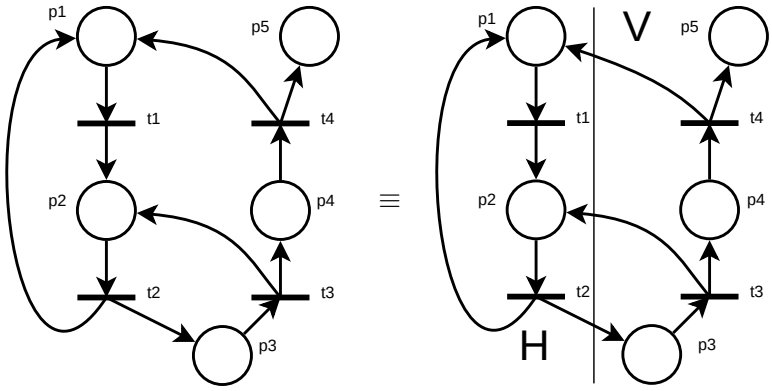


FIGURE 3.6: Selecting subnet to hide

Example 3.7. *Consider the Petri net of the figure 3.6 with the next incidence matrix:*

$$\begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{pmatrix} -1 & 1 & 0 & 1 \\ 1 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

The subnet we want to hide is formed by places p_1 and p_2 and transitions t_1 and t_2. Graphically, separate places and transitions to hide (H) from the rest of the network (V)

The incidence matrix is already sorted by the places and transitions to the top of it. Here are the four parts described above.

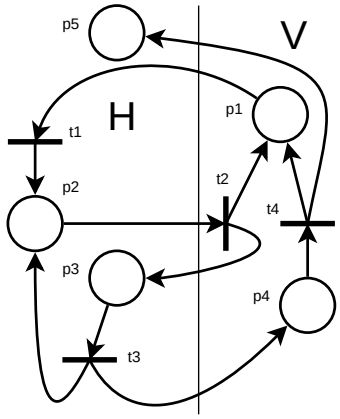
$$\begin{array}{c|cc|cc} & t_1 & t_2 & t_3 & t_4 \\ p_1 & -1 & 1 & 0 & 1 \\ p_2 & 1 & -1 & 1 & 0 \\ \hline p_3 & 0 & 1 & -1 & 0 \\ p_4 & 0 & 0 & 1 & -1 \\ p_5 & 0 & 0 & 0 & 1 \end{array}$$

In this matrix we can see:

- $H = \begin{array}{c} \\ p_1 \\ p_2 \end{array} \begin{array}{c} \begin{array}{cc} t_1 & t_2 \end{array} \\ \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} \end{array}$ *is the subnet we want to hide. It is the same as* N_1
- $V = \begin{array}{c} \\ p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{cc} t_3 & t_4 \end{array} \\ \begin{pmatrix} -1 & 0 \\ 1 & -1 \\ 0 & 1 \end{pmatrix} \end{array}$ *is the subnet that is visible. It is the same as* N_2
- $HP = \begin{array}{c} \\ p_1 \\ p_2 \end{array} \begin{array}{c} \begin{array}{cc} t_3 & t_4 \end{array} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{array}$ *are the relationships between transitions of* V *and* H *places. It is the equivalent to* PIM_1 *or* TIM_2
- $HT = \begin{array}{c} \\ p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{cc} t_1 & t_2 \end{array} \\ \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \end{array}$ *are the relationships between places of* V *and* H *transitions. It is the equivalent to* TIM_1 *or* PIM_2

At this moment, it is easy to see that we can choose any subset of places and transitions in order to hide it, simply reordering rows and columns.

Example 3.8. *In the previous example we have seen a fairly simple option selection subnet and we have chosen the place* p_1 *and* p_2 *and transitions* t_1 *and* t_2*. However, we can choose any other subset of places and transitions. In this example we will select places* p_2*,* p_3 *and* p_5 *and the transitions* t_1 *and* t_3*. Thus, in the graph of the previous example move the locations and transitions to hide on one side and the rest on the other.*



Although more confusing, can be seen that the graph is the same as the incidence matrix is the same (not just part of the equivalence class, it is exactly the same). Now, in this matrix move places p_2, p_3 and p_5, and transitions t_1 and t_3 at the beginning of the matrix:

$$\begin{array}{c|cc|cc} & t_1 & t_3 & t_2 & t_4 \\ p_2 & 1 & 1 & -1 & 0 \\ p_3 & 0 & -1 & 1 & 0 \\ p_5 & 0 & 0 & 0 & 1 \\ \hline p_1 & -1 & 0 & 1 & 1 \\ p_4 & 0 & 1 & 0 & -1 \end{array}$$

Interpreting each of the chunks of the matrix is similar to the previous example.

3.2.5 Front-end interaction with the subnet. Input and output functions

Once you have defined all this environment, we will try to go a little further. Let's assume that we want to export a subnet we have hidden in another network, like a black box. Our intention is to connect this hidden network to another network, and can thus be reused subnets. For example, let's assume that we have a process modeling with Petri net modeling and in this there is a subnet we want to hide, but, at the same time, we want to reuse it in other Petri nets.

In this case we have a problem, and once hidden network disappears half the information input or output arcs of the same. In particular, we do not know the source nodes and arcs that leave the target nodes of the arcs that enter the network until no visible again. But if we want to reuse it on other networks, can not wait to make it visible. Should remain hidden, but should be able to connect to other networks.

We will try to solve this problem. This way we can reuse hidden networks like plug-in modules on other networks. However, we will not need the actual implementation of the source or destination nodes of the arcs that leave or enter the network, respectively. The solution is to define a facade or front-end input and output of the network. This front-end will contain the information needed to interact with the network hidden, but hide the specifics of implementation. To define this behavior going from some assumptions.

3.2.5.1 Previous definitions

Let $N = \langle P, T, \alpha, \beta \rangle$ be a Petri net and let $P = \{N_1, N_2\}$ be a partition of N.

Definition 3.11 (Input place). Let p_i a place of N_1. p_i is an input place of N_1 if it is the destination of an arc coming from a N_2 transition, ie,

$$p_i \text{ is an input place of } N_1 \text{ if } \exists t_j \in N_2 \ | c_{ij} > 0$$

Definition 3.12 (Input transition). Let t_i a transition of N_1. t_i is an input transition of N_1 if it is the destination of an arc coming from a N_2 place, ie,

$$t_i \text{ is an input transition of } N_1 \text{ if } \exists p_j \in N_2 | c_{ji} < 0$$

Definition 3.13 (Input node). An input node of N_1 is an input place or transition of N_1.

Definition 3.14 (Output place). Let p_i be a place of N_1. p_i is an output place of N_1 if an arc leaves it towards a transition of N_2, ie,

$$p_i \text{is an output place of } N_1 \text{ if } \exists t_j \in N_2 | c_{ij} < 0$$

Definition 3.15 (Output transition). let t_i be a transition of N_1. t_i is an output transition of N_1 if an arc leaves it towards a place of N_2, ie,

$$t_i \text{ is an output transition of } N_1 \text{ if } \exists p_j \in N_2 | c_{ji} > 0$$

Definition 3.16 (Output node). An output node of N_1 is an output place or transition of N_1.

After defining these concepts, we can define the sets thereof.

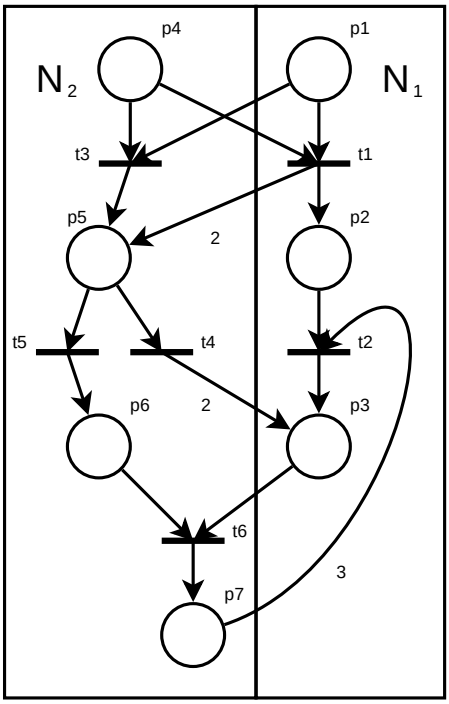
Notation. We denote the sets of the elements defined above:

- Let $IP(N) \subseteq \overline{P}$ (Input Places) be the set of input places of a subnet.
- Let $IT(N) \subseteq \overline{T}$ (Input Transitions) be the set of input transitions of a subnet.

- Let $IN(N) \subseteq \overline{P} \cup \overline{T}$ (Input Nodes) be the set of input nodes of a subnet.
- Let $OP(N) \subseteq \overline{P}$ (Output Places) the set of output places of a subnet.
- Let $OT(N) \subseteq \overline{T}$ (Output Transitions) be the set of output transitions of a subnet.
- Let $ON(N) \subseteq \overline{P} \cup \overline{T}$ (Output Nodes) be the set of output nodes of a subnet.

Note. Recall that a node in a Petri net can be both a place and a transition, depending on the context.

Notation. Denote as n_i to a node of a Petri net.



$$C = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ \hline p_4 \\ p_5 \\ p_6 \\ p_7 \end{array} \begin{array}{c} \begin{array}{cc} t_1 & t_2 \end{array} \\ \left(\begin{array}{cc} -1 & 0 \\ 1 & -1 \\ 0 & 1 \\ \hline -1 & 0 \\ 2 & 0 \\ 0 & 0 \\ 0 & -3 \end{array}\right. \end{array} \begin{array}{c} \begin{array}{cccc} t_3 & t_4 & t_5 & t_6 \end{array} \\ \left.\begin{array}{cccc} -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & -1 \\ \hline -1 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{array}\right) \end{array}$$

FIGURE 3.7: Subnets with input and output nodes

As we have generic definitions, no problem in applying to a network divided into H, V, HN and HT, as the set $\{H, V\}$ is a partition of N.

3.2.5.2 Subnet Front-end

Once all these concepts, we create the front-end input/output of a Petri subnet. A front-end of the Petri net will be a intermediate facade that allows us to physically divide that subnet from the rest of the net. Thus, in order to enter or leave the subnet, you need to make it through this front-end.

Let IA (input arcs) the set of arcs that enter the subnet N_1 and let OA (output arcs) the set of arcs leaving N_1.

Definition 3.17 (Input gate of a net). Let $a_i \in IA$ an arc of entrance to N_1. We define an input gate to N_1, and denote by ig_i, as a new logical node that is identified with an arc of entrance to the net. For each input arc, defines an input gate with the same weight, regardless of the origin and destination of the arc. If the source is a transition, we denote igt_i and if a place, igp_i.

Definition 3.18 (Output gate of a net). Let $a_i \in OA$ output arc N_1. We define an output gate of N_1, and denote by og_i, as a new logical node that is identified with an exit arc of the net. For each exit arc is defined an output gate with the same weight, regardless of the origin and destination of the arc. If the source is a transition, we denote ogt_i and if it is a place, ogp_i.

In this way we can divide the input arcs and output into two parts: a N_1 internal and external to N_1. If we take an arc of entrance a_i that has an origin in n_j and destination in n_k, we define an input gate through a point of entry so that the original arc a_i is divided into two parts.

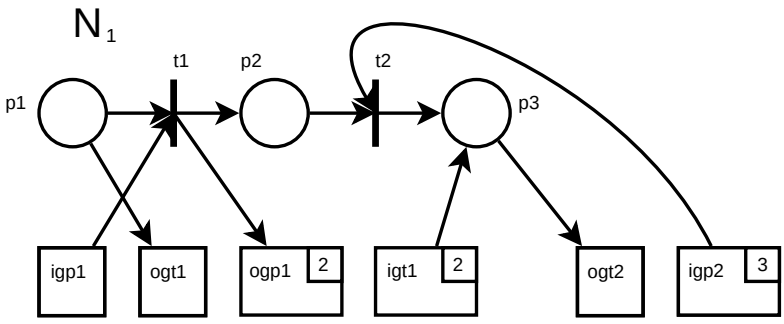
- a_{i1} (external to N_1) with origin in n_j and destination in igt_i or igp_i depending on if n_j is a transition or a place, and the same weight as the input gate.
- a_{i2} (internal to N_1) with destination in igt_i or igp_i depending on if n_j is a transition or a place respectively, and the same weight as the input gate.

Similarly, if we take an exit arc a_i that has an origin in n_k and destination in n_j, we define an output gate og_i so that the original arc a_i is divided into two parts:

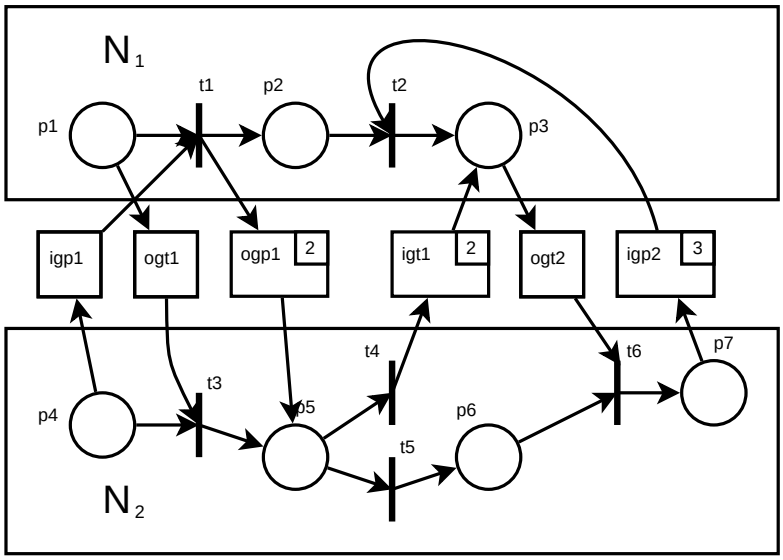
- a_{i1} (internal to N_1) with origin in n_k and destination in ogt_i or ogt_i depending on if n_j is a transition or a place, and the same weight as the output gate.
- a_{i2} (external to N_1) with destination in n_j and origin in igt_i or igp_i depending on if n_j is a transition or a place respectively, and the same weight as the output gate.

Example 3.9. *Consider the net in figure 3.7. In this network we have three arcs entering and three arcs leaving. For each of those that leave, I define output gates and for each coming one, we define input gates. The weight of each gate (if it is different than 1) is*

placed in a little square inside. The subnet N_1 becomes:



and in the complete net, arcs entering and leaving are divided into two pieces:



Definition 3.19 (Input Front-end of a net). The input front-end (or input interface) of a subnet N_1 is the set of all input gates of N_1. We denote by IF of N_1.

Definition 3.20 (Output Front-end of a net). The output front-end (or output interface)of a subnet N_1 is the set of all output gates of N_1. We denote by OF of N_1.

Definition 3.21 (Front-end of a net). The front-end (or interface) of a net N_1 is the pair of IF and OF of N_1. We denote by F of N_1.

$$F = \langle IF, OF \rangle$$

Example 3.10. *Taking the net of the example 3.9 and applying these new definitions, we would have N_1 net along with its front end as shown in Figure 3.8.*

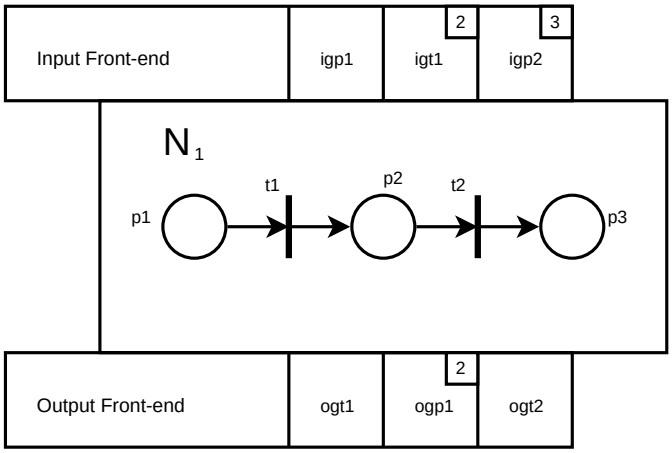


FIGURE 3.8: Front-end of a net

3.2.5.3 Input/output functions

Once all these input and output concepts defined, we will introduce a few key concepts for our purpose.

Let N a Petri net and let $\{N_1, N_2\}$ a partition of N. Let $F = \langle IF, OF \rangle$ the front-end of N_1.

Definition 3.22 (Petri net Input function). We define the input function f_i of N_1 as:

$$f_i : F \longrightarrow IN$$

such that for each input gate igt_i you mapped one or no input place N_1 and each input gate igp_j you mapped one or no input transition N_1.

Definition 3.23 (Petri net Output function). We define the output function f_o of N_1 as:

$$f_o : ON \longrightarrow F$$

such that each output place N_1 you mapped one or no output gate ogt_i of N_1 and each output transition N_1 you mapped one or no output gate ogp_j to N_1

The input function can be defined for all the input gates and the output function should be surjective because if not, some door would not be connected. Anyway that is not essential. If a front-end door is not connected with any element of your network, simply by solving the final network, the arcs connected to that door disappear. Note also that the input function is not necessarily injective: Multiple input gates can be associated to the same node of N_1.

Example 3.11. *Consider the net* N_1 *in figure 3.7 with its front-end in figure 3.8. The input and output functions are:*

- *Input function:*

F	igp_1	igt_1	igp_2
IN	t_1	p_3	t_2

- *Output function:*

ON	p_1	t_1	p_3
F	ogt_1	ogp_1	ogt_2

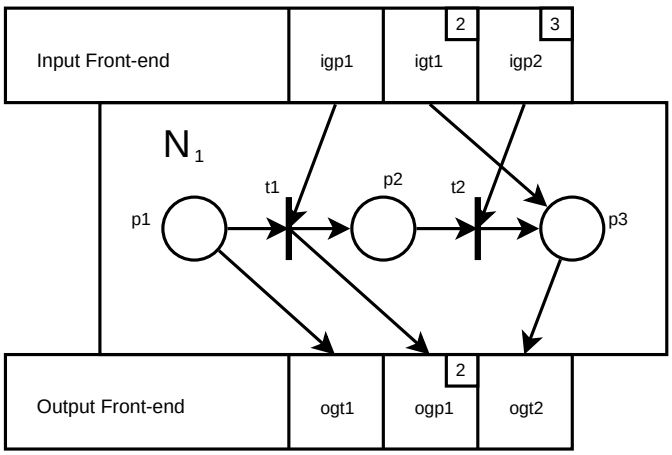
3.2.5.4 Attachable net

By joining the subnet R_1 along with its front-end and its input and output functions f_i and f_o we grouped both the internal network with external communication. This way we can "extract" a subnet and "implant" it in another net. You only need this destination network is to communicate with the front-end. So naturally appears the following definition.

Definition 3.24 (Attachable Petri net). An Attachable Petri net is a quadruple $R_a = \langle R, F, f_i, f_o \rangle$

From these definitions, it is clear that you can create attachable subnets taking a subnet of another given and applying the whole process we have defined. But it is also possible to create from scratch, starting from a network, defining a front end for that network and declaring the input and output functions. So you can create Petri nets modules providing functionality and out through a front-end without requiring the actual implementation.

Example 3.12. *The attachable net in figure 3.7 would be the next:*

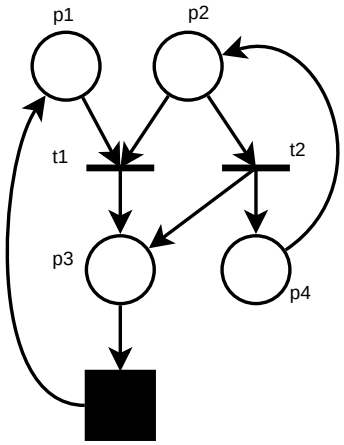


It can be seen as a private black box with visible input and output connectors that are "plugged" to other networks. In a attachable net, the private part would be N_1, f_i

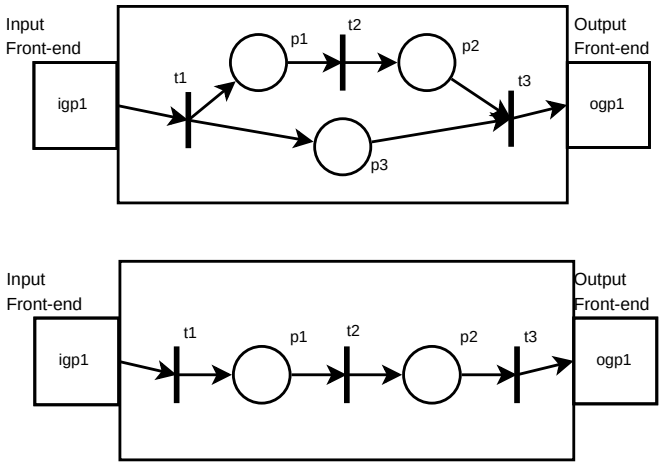
and f_o. The public part of the front-end would be F. All a net need to know is the input/output front-end.

A utility of these nets is that its definition is simple, since only the front-end is needed to define its operation. This makes possible to create nets using attachable nets in certain areas where they do not know their actual implementation, but its behavior. Additionally, it is possible to use different implementations of "network providers" of the same attachable nets, using at each moment the most appropriate one.

Example 3.13. *Consider now the following Petri net*



to which we want to connect an attachable net in the black box. Let's assume we have two equivalent alternatives described in Example 3.16:



We Can "plug" either because their front ends are equivalent and remains in figure 3.9.

In this case, the behavior of the net will be the same, but does not have to be. That will decide who connects nets. For example, you could create a "foo" net that does nothing at first and replace it later by the real one.

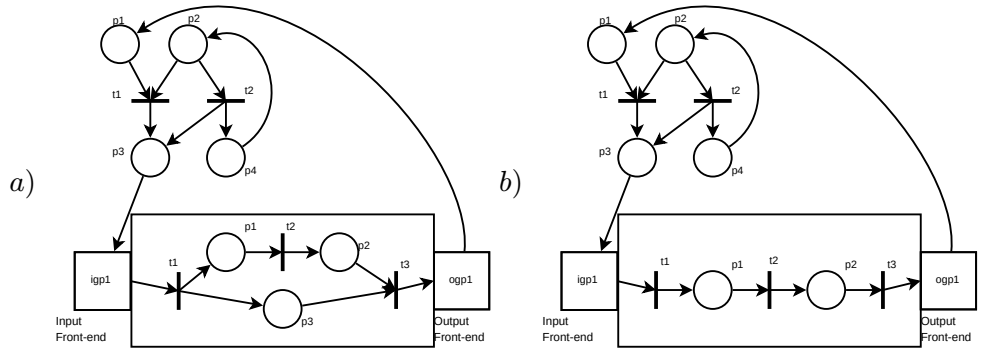


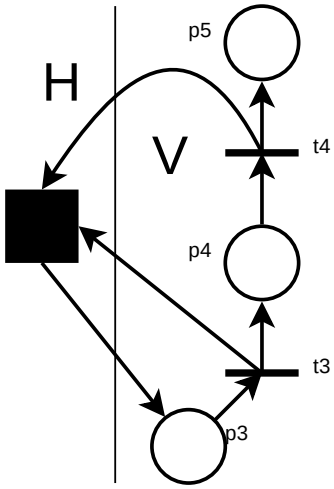
FIGURE 3.9: Two different implementations of attachable nets

3.3 Private information. Hiding a subnet

Once we have defined subnets and front-ends we can select one of them to securize it by hiding. We proceed to the occultation as such [4]. Graphically, it seems simple. Just replace the subnet to hide by a black box and modify some arcs according to the following rules:

1. The arcs originating in a place or transition within the black box, and target a place or transition out of it will have the black box as the source.
2. The arcs originating in a place or transition out of the black box, and target a place or transition within it, are replaced by the black box as a destination.

Example 3.14. *We consider the Petri net of the Figure 3.6. The result of hiding the part of the graph H is the following:*



In the associated incidence matrix also replace the H elements by a black box:

	t_1	t_2	t_3	t_4
p_1	■	■	0	1
p_2	■	■	1	0
p_3	0	1	-1	0
p_4	0	0	1	-1
p_5	0	0	0	1

However, in this matrix notation is given information should also be hidden: it gives us information about the number of places and transitions of the hidden subnet, besides indicating hidden places and transitions with which it interacts. To solve this problem we proceed as follows. We can group all rows for the screened subnet into one. In each row position examine all elements of the original rows corresponding to that position, and will put:

- If all these elements are zero, in the grouped row will be a zero.
- If one and only one of those elements is nonzero, will put that item.
- If there are several non-zero elements, we will post a list of these items separated by commas, creating a d-dimensional element (in d dimensions).

In the same way we have done with the rows, proceed with columns. Thus, if the hidden subnet has i columns and j rows, we will get a matrix like this:

■	$a_{1(i+1)}$	$\cdots$	a_{1m}
$a_{(j+1)1}$			
$\vdots$		V	
a_{n1}			

where $\forall p, \forall q | i+1 \leq p \leq m \wedge j+1 \leq q \leq n$

$$a_{1p} = \begin{cases} 0 & \text{if } \forall r | 1 \leq r \leq j, c_{rp} = 0 \\ c_{rp} & \text{if } \exists! r, 1 \leq r \leq j | c_{rp} \neq 0 \\ (c_{r_1 p}, c_{r_2 p}, ...) & \text{if } \exists r_1 \neq r_2 \neq ..., 1 \leq r_1, r_2, ... \leq j | c_{r_1 p}, c_{r_2 p}, ... \neq 0 \end{cases}$$

$$a_{q1} = \begin{cases} 0 & \text{if } \forall s | 1 \leq s \leq i, c_{qs} = 0 \\ c_{qs} & \text{if } \exists! s, 1 \leq s \leq i | c_{qs} \neq 0 \\ (c_{qs_1}, c_{qs_2}, ...) & \text{if } \exists s_1 \neq s_2 \neq ..., 1 \leq s_1, s_2, ... \leq s | c_{qs_1}, c_{qs_2}, ... \neq 0 \end{cases}$$

So we hide the number of places and transitions of the hidden subnet and their relationships. Yes, some information is given about the hidden network. Really if this resulting matrix some node that is d-dimensional, at least in the hidden network must exist d nodes of this type.

Furthermore, each nonzero element define part of the front-end of the subnet. If we take the elements of the first row (the first element of the column is the hidden subnet):

- any element greater than zero defines an *ogt* towards the transition related by the column
- any element less than zero defines an *igt* from the transition related by the column

If we take the elements of the first column (the first element of the column is the hidden subnet):

- any element greater than zero defines an *igp* towards the place related by the row
- any element less than zero defines an *ogp* from the place related by the row

With this information we can build the entire front-end of the subnet.

Example 3.15. *We consider the Petri net defined by the following incidence matrix, separated into H,V,HT and HP.*

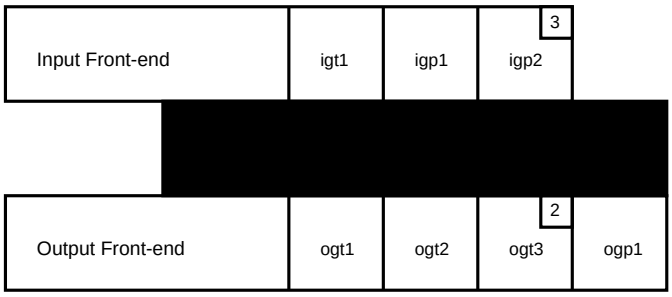
$$\begin{array}{c|ccc|ccc} & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 \\ p_1 & -1 & 0 & 1 & 0 & 0 & 0 \\ p_2 & -1 & 0 & 0 & 1 & 0 & 1 \\ p_3 & 1 & 0 & 0 & -1 & 0 & 0 \\ p_4 & 1 & -1 & 0 & 2 & 0 & 0 \\ \hline p_5 & 0 & 1 & 0 & 0 & 0 & 0 \\ p_6 & 3 & 0 & -1 & 0 & 0 & 0 \\ p_7 & 0 & 0 & 0 & -1 & -1 & 1 \\ p_8 & 0 & 0 & 0 & 0 & 1 & -1 \end{array}$$

After applying the above steps for the group, we would have the following:

	■	t_4	t_5	t_6
	■	$(1,-1,2)$	0	1
p_5	1	0	0	0
p_6	$(3,-1)$	0	0	0
p_7	0	-1	-1	1
p_8	0	0	1	-1

Here we see that the information about the number of hidden places and transitions is minimized. So we know that at least there are two hidden transitions and at least three hidden places (there is a transition of dimension 3). However, we do not know the exact number of either.

Now we can build the front-end. The elements of the first row are $(1,-1,2)$ *and* 1*. So we have three ogt and one igt. The elements of the first column are* 1 *and* $(3,-1)$*, so defines two igp and one ogp like this. This is the subnet front-end, having hidden its inner content.*



3.3.1 Hiding vs. Reduction

Both Silva works [12, 15] as in the article by Xia [10] discusses possible Petri nets reductions for grouping and simplifying, under certain circumstances, places and/or transitions. These reductions can be structural (only dependent on the structure and initial marking of the net) or depending on the interpretation of the Petri net.

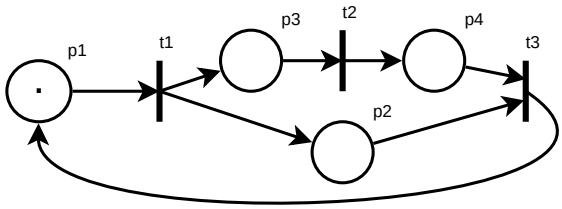
Should be clear that these reductions are not the same thing we are describing. We do not try to simplify the network together elements to have more or fewer places or transitions or to make it easier. What we want is to hide part of the network, regardless of how simple or complicated it is.

Here we have an example of what a reduction is.

Example 3.16 (Reduction of an implicit place [12]). *In a marked Petri net, an implicit place is one that meets the following:*

1. *its marking can be calculated from other points marking*
2. *never is the only place that prevents the enabling of its output transitions*

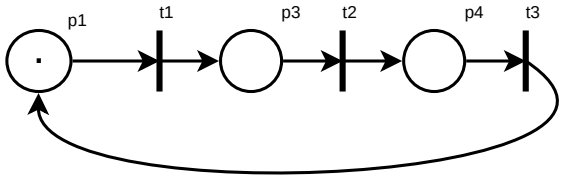
If we consider the following Petri net



we can notice that p_2 is an implicit place because its marking can be calculated as a function of p_3 y p_4:

$$M(p_2) = M(p_3) + M(p_4)$$

Moreover, by this same formula, it is clear that $M(p_2) \geq M(P_4)$ (marking cannot be negative) so the only place that can prevent enabling of T_3 is P_4. Thus eliminating p_2 does not alter the behavior of the network, which would be as follows:



In this network elements have been removed, no hidden. This example helps us to see the difference between hiding and a reduction.

3.4 Conclusions

As we have seen, any Petri net can be divided into any number of subnets, only limited by the number of places and transitions. Furthermore, each one of this Petri subnets has its own input and output front-ends in order to connect with the rest of the Petri net. Of course, the main application of this definitions in this thesis is that the private information of this Petri net is stored in one or more of these defined subnets.

So I have reached the first milestone: to extend Petri Nets definition in order to define the public and the private information.

Chapter 4

Petri net representation for subnets support. PNML

4.1 Introduction

Once explained how to split a net in several subnets, the next step is to define a way to secure one or more of those subnets.

There is not literature about this topic. Because of that I have to do a previous work about Petri net representation. First of all a way to represent subnet must be defined. Depending on the selected representation, the way to occult subnets may be different or even impossible.

4.2 Petri net representations

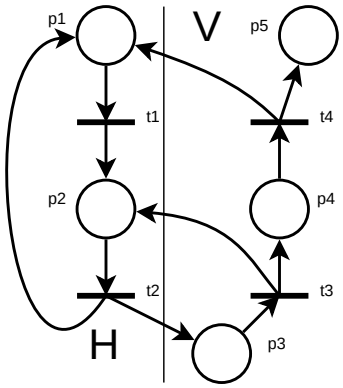
There are four standard ways to represent Petri nets. Each one of them have their properties, advantages and disadvantages. But I want to select one that I am able to represent any kind of Petri net, its subnets and allow to hide information without erasing it.

4.2.1 Graphic representation

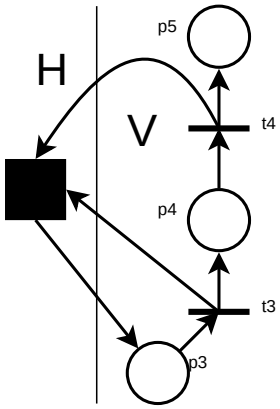
This is the clearest and extended way to represent Petri nets. It has a very important advantage and it is that a picture is worth a thousand words.

Subnets can be defined simply drawing a vertical line. The right part is one subnet and the left part is other subnet. Places and transitions can be moved from one location to another depending on the subnet they are situated. This is only an example. Other way would be to use colors for the nodes (same color indicates same subnet) or use rectangles, etc.

Example 4.1 (Graphic representation of a hidden subnet). *Let's take the following Petri net.*



I want to occult the part marked with an H, so the result is this graphic



And now, how can I maintain the H subnet information on the black box? If I want to distribute this Petri net I should send something more to that people I want to access the hidden subnet. I have not found a way to embed that information in the graphic so that some people can access it but other people can't.

So this representation is useful in order to show at one sight the Petri net structure, but I can't choose it for my goals. I have not been able to discover a way to show some people the hidden information (the hidden subnet). However I will continue using it where a clear idea of the Petri structure if necessary.

4.2.2 Matrix representation

This representation is very useful to study properties and evolution of a Petri net, independently of its graphic representation. As we have seen before (in chapter 3), we can reorder rows and columns and define subnets in the matrix.

Example 4.2 (Matrix representation of a hidden subnet). *This is the matrix representation of the Petri net in the previous example 4.1.*

$$\begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{pmatrix} -1 & 1 & 0 & 1 \\ 1 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

As we can see before, If I occult the part marked with an H, so the result is:

$$\begin{array}{c} \\ p_1 \\ p_2 \\ \hline p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{cc|cc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \left(\begin{array}{cc|cc} \blacksquare & \blacksquare & 0 & 1 \\ \blacksquare & \blacksquare & 1 & 0 \\ \hline 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{array}\right) \end{array}$$

or this other one, grouping the places and transitions of the hidden subnet as seen in chapter 3

$$\begin{array}{c} \\ \\ p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{c|cc} & t_3 & t_4 \end{array} \\ \left(\begin{array}{c|cc} \blacksquare & 1 & 1 \\ \hline 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{array}\right) \end{array}$$

And now I have the same problems as in graphic mode: how can I keep information in the black box?

With this representation it is possible to study properties and it may be really important as a complement to graphic mode. With both representations together, everyone has a clear idea of the Petri net structure and properties. But I haven't found a way to store information inside the black box.

4.2.3 Equation representation

The third representation way for Petri nets is the equation representation. Basically, transitions are selected and, for each one, the tokens of the places connected to that transition are modified. This is very useful to compute the evolution of a Petri net, choosing the transition fired.

However it is difficult to find a way for representing subnets with this notation. And, of course, if a subnet cannot be represented, it cannot be hidden.

Example 4.3 (Equation representation of a hidden subnet). *Let's take again the Petri net from the previous examples 4.1. This is It's equation representation:*

```
if (p1>0) then
        p1 <- p1 - 1
        p2 <- p2 + 1
if (p2>0) then
        p2 <- p2 - 1
        p1 <- p1 + 1
        p3 <- p3 + 1
if (p3>0) then
        p3 <- p3 - 1
        p2 <- p2 + 1
        p4 <- p4 + 1
if (p4>0) then
        p4 <- p4 - 1
        p1 <- p1 + 1
        p5 <- p5 + 1
```

As we can see, we should hide several lines. In particular all that includes places p_1 and p_2, resulting something like this (strikethrough text would be part of the subnet):

```
if (p1>0) then
        p1 <- p1 - 1
        p2 <- p2 + 1
if (p2>0) then
        p2 <- p2 - 1
        p1 <- p1 + 1
        p3 <- p3 + 1
if (p3>0) then
        p3 <- p3 - 1
        p2 <- p2 + 1
        p4 <- p4 + 1
if (p4>0) then
        p4 <- p4 - 1
        p1 <- p1 + 1
        p5 <- p5 + 1
```

This is probably the strangest way to represent Petri nets and it is difficult to define subnets over it.

I have tried to think about these ways of representation but, in my opinion, no one of them is suitable enough to represent subnets in a clear way than can be occulted. Because of that I have chosen the fourth representation, which is PNML, and that is explained in the next section.

4.3 PNML. Petri Net Marked Language

As I have explain in the Literature review in the chapter 2, PNML is a way to represent Petri nets as xml content.

The advantages over the other three representation ways described before are clear. By one side, XML is a widely extended format to represent almost everything. And not only that. XML is a robust technology free of errors and it is really flexible. Its flexibility come from you can add any kind of labels and functionality with a very little amount of work. Its robustness come from the strict specification of the schemas declared to define completely the XML files that support. Once the schema is defined, the XML files have a no way to get out of this definition, so we can know if a XML file is correct given its schema. And this is not all. If you have an XML file, you can extract information to and complete it to create a schema for that file.

Originally, with basic Petri nets, the structure of a Petri net was fully provided. The only thing that is not supported in comparison with graphic mode is the graphical appearance of Petri nets: the position of nodes and transitions was not important, but with the arrival of High level Petri nets and Petri nets design software, it is necessary to store this kind of information.

In this work, I am going to study only basic Petri nets, but that concept and the method is easily exportable to other kind of nets, such as Symmetric nets and High Level Petri nets (that are representable in PNML format too.)

4.3.1 Scope

The scope of this work in this section is basically bounded by the original and basic Petri nets, that is:

- there are places, transitions and arcs between them

- places can have tokens on them (but this does not influence the hiding process)
- transitions can be fired, but there is only one type of transition (not messages, not time, etc.)

Furthermore, I am going to explain several concepts for a specific graphical design of the Petri net (that will not influence the process either).

There are other options such as specific information for the design tool that I am not going to discuss because the tools have no interest in this work

4.3.2 Description

Petri Net Marked Language is an xml language created to represent Petri Nets. With this language we can take a Petri Net and store it into an xml file without loss of information.

One of the best properties of PNML is that, as it is an xml based schema, it can be extended with more functionality extending the grammar. Virtually, any extension over Petri nets can be translated into PNML in a logical and natural way. Moreover, this extension is defined by Petri net type definition [66, 67].

In this case PNML hasn't got a way to represent subnets. There is something named `<page>` that is used to represent several nets in the same PNML file. But, by default, a node inside a page cannot connect with a node of other page. So it cannot be used as "subnets". So I am going to extend the language in order to get several goals:

1. Represent subnets of a Petri Net.
2. Include input and output interfaces for every subnet.

As we can think, definition of several subnets of a Petri net is possible and the connection over them are always through their respective interfaces.

4.3.3 PNML grammar

As PNML is an xml based language it has to be described by an schema that define the creation rules of the PNML representation of a Petri net.

The grammar is defined since 2009 and updated until 2012, which is the most recent revision. I am not going to do and extensive explanation of all the possibilities of the

grammar, but the most important. As we can see later, anything we think is useful can be added to the process with little effort. So I am going to study only the most basic elements of a Petri net. The rest of the element can be attached later with facility.

4.3.3.1 PNML basics

In this section I am going to explain several characteristics of PNML files. With these explanations it is going to be easier the understanding of PNML structure.

First of all, as PNML files are xml files, there several things to know:

1. A xml file normally starts with a line defining some characteristics of the file, like the version and the encoding type. It has an aspect like this[1]:

```
<?xml version="1.0" encoding="utf-8"?>
```

2. A root node must exist. In this case, the root node is `<pnml>`. So every PNML files has to start with the tag `<pnml>` and end with the tag `</pnml>`. Below this tag, there is a new tag `<net>` that can contain

   - Type: the type of the Petri net as an attribute. In this case, as I am going to study only Place/Transition nets, it will be `ptnet`
   - Name of the net: New tag `<name><text>...</text></name>`
   - Pages: one page is an invention to store several Petri nets inside an unique PNML file, but usually there is only one page for file. It is nested inside a `<page>` tag.

3. Each element in PNML has to have a unique id inside the net to be identified unambiguously. So there cannot be two elements with the same id.

With this three observations, we can have an idea about how a PNML file is structured:

```
<?xml version="1.0" encoding="utf-8"?>
<pnml>
  <net id="myNet" type="http://www.pnml.org/version-2009/grammar/ptnet">
    <name>
      <text> My new net </text>
    </name>
    <page id="page1">
      .......
    </page>
  </net>
</pnml>
```

[1]For clarity, in the following examples, this line can be deleted.

LISTING 4.1: Example of general PNML file

Once this structure is defined, I am going to explain the next stage, that is the most important one in this work.

4.3.3.2 Places, transitions and arcs in PNML

As Petri nets has three main elements, places, transitions and arcs, and PNML has them too. These three elements have several things in common in an PNML file:

- They are all nested in a page tag
- Places and transitions (not arcs) can contain a tag `<name>` with its name. This tag has been defined before for the name of the net. It can store information about the text of the name and the graphical position of this label in this way:

```
<name>
  <text> Element Name </text>
  <graphics>
    <offset x="22" y="-10"/>
  </graphics>
</name>
```

- They can contain a tag `<graphics>` with information about its position and dimension:

```
<graphics>
  <position x="100" y="200"/>
  <dimension x="40" y="40"/>
</graphics>
```

These are the common properties of places, transitions and arcs. Now let's go on the particular characteristics of each one of them.

Places are represented with the tag `<place>` and the can have a marking with the label `<initialMarking>`. Here we have an example of a place with two tokens in PNML:

```
<place id="p1">
  <name>
    <text> Place number one </text>
    <graphics>
      <offset x="130" y="130"/>
    </graphics>
  </name>
  <graphics>
```

```
    <position x="130" y="90"/>
    <dimension x="40" y="40"/>
  </graphics>
  <initialMarking>
    <text> 2 </text>
  </initialMarking>
</place>
```

Listing 4.2: PNML representation for places

Transitions are represented with the tag `<transition>`. This is an example of a transition in PNML:

```
<transition id="t1">
  <name>
    <text> Transition number one </text>
    <graphics>
      <offset x="270" y="140"/>
    </graphics>
  </name>
  <graphics>
    <position x="270" y="100"/>
    <dimension x="40" y="40"/>
  </graphics>
</transition>
```

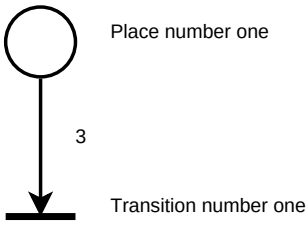
Listing 4.3: PNML representation for transitions

Arcs are represented with the tag `<arc>`. Arc have to have a source and a target, which are defined by the attributes `source` and `target` that have to point to a transition and a place, identified by their id. Furthermore, the arc weight can be fixed by the `<inscription>` tag. If the weight is one, the tag inscription is not necessary because this is the default value. This is an example of the arc with weight 3 that connects the place and the transition of the previous examples in PNML:

```
<arc id="a1" source="p1" target="t1">
  <inscription> 3 </inscription>
</arc>
```

Listing 4.4: PNML representation for arcs

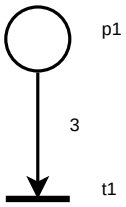
If we take the last examples all together, we can represent the following Petri net:




```
<?xml version="1.0" encoding="utf-8"?>
<pnml>
  <net id="myNet" type="http://www.pnml.org/version-2009/grammar/ptnet">
    <name>
      <text> My new net </text>
    </name>
    <page id="page1">
      <place id="p1">
        <name>
          <text> Place number one </text>
          <graphics>
            <offset x="130" y="130"/>
          </graphics>
        </name>
        <graphics>
          <position x="130" y="90"/>
          <dimension x="40" y="40"/>
        </graphics>
        <initialMarking>
          <text> 2 </text>
        </initialMarking>
      </place>
      <transition id="t1">
        <name>
          <text> Transition number one </text>
          <graphics>
            <offset x="270" y="140"/>
          </graphics>
        </name>
        <graphics>
          <position x="270" y="100"/>
          <dimension x="40" y="40"/>
        </graphics>
      </transition>
      <arc id="a1" source="p1" target="t1">
        <inscription> 3 </inscription>
      </arc>
    </page>
  </net>
</pnml>
```

Listing 4.5: Complete PNML representation for a basic Petri net

For clarity, I am going to obviate several options. I am not going to put the graphic information and the names, but the id. So this last example is as follows:




```
<?xml version="1.0" encoding="utf-8"?>
<pnml>
  <net id="myNet" type="http://www.pnml.org/version-2009/grammar/ptnet">
    <page id="page1">
      <place id="p1">
        <initialMarking>
          <text> 2 </text>
        </initialMarking>
      </place>
      <transition id="t1"/>
      <arc id="a1" source="p1" target="t1">
        <inscription> 3 </inscription>
      </arc>
    </page>
  </net>
</pnml>
```

Listing 4.6: Basic PNML representation for a basic Petri net

I think that this last listing is clear enough to understand the rest of the process. But not only that. For even more clarity, the tags `<?xml>`, `<pnml>`, `<net>` and `<page>` are going to be obviated too. So in many of the later examples they will not be present. Applying this criterium, the listing 4.6 change to:

```
<place id="p1">
  <initialMarking>
    <text> 2 </text>
 </initialMarking>
</place>
<transition id="t1"/>
<arc id="a1" source="p1" target="t1">
  <inscription> 3 </inscription>
</arc>
```

Listing 4.7: Simplified PNML representation for a basic Petri net

4.3.4 PNML examples

In this section I will present several examples of Petri nets represented with PNML. This examples are extracted directly form www.pnml.org. These examples will be processed later in order to hide parts of them.

As first example, we have the Dining Philosophers. It is a classic problem solved by this well known Petri net:

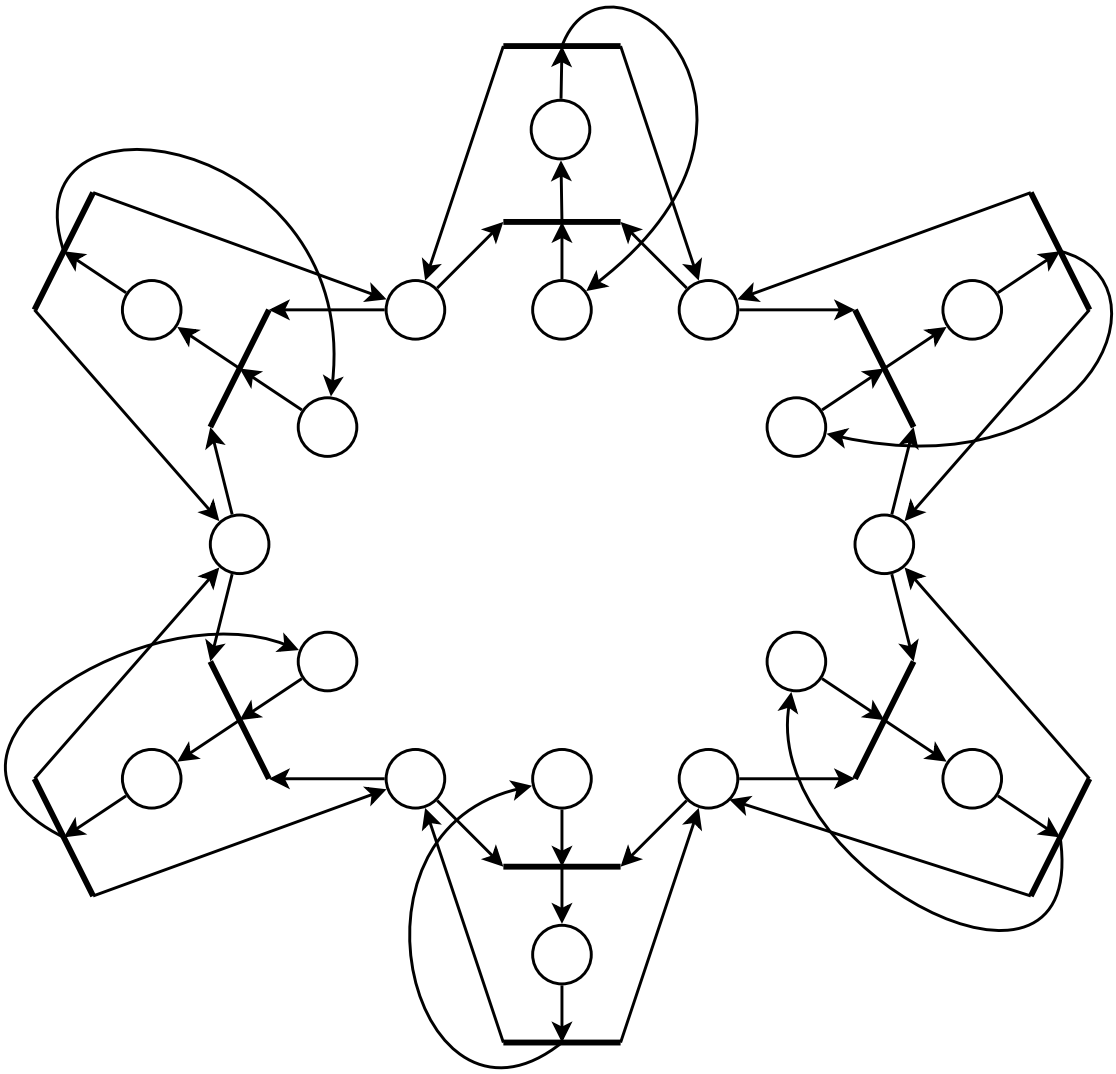


FIGURE 4.1: Dining philosophers Petri net

Obviating the graphics information about the exact position of the elements, the PNML file that represents this Petri net is:

```
<pnml xmlns="http://www.pnml.org/version-2009/grammar/pnml">
 <net id="i943123747" type="http://www.pnml.org/version-2009/grammar/ptnet">
  <name>
   <text>philo</text>
  </name>
  <page id="i-1410027568">
   <place id="cId175">
    <name>
     <text>FORK_1</text>
    </name>
    <initialMarking>
     <text>1</text>
    </initialMarking>
   </place>
   <place id="cId169">
    <name>
     <text>WAIT_RIGHT_FORK_2</text>
    </name>
   </place>
   <place id="cId171">
    <name>
     <text>FORK_3</text>
    </name>
    <initialMarking>
     <text>1</text>
    </initialMarking>
   </place>
   <place id="cId172">
    <name>
     <text>WAIT_RIGHT_FORK_5</text>
    </name>
   </place>
   <place id="cId165">
    <name>
     <text>EAT_5</text>
    </name>
   </place>
   <place id="cId162">
    <name>
     <text>THINK_4</text>
    </name>
    <initialMarking>
     <text>1</text>
    </initialMarking>
   </place>
   <place id="cId177">
    <name>
     <text>EAT_4</text>
    </name>
   </place>
   <place id="cId166">
    <name>
     <text>WAIT_LEFT_FORK_3</text>
    </name>
   </place>
   <place id="cId161">
    <name>
     <text>WAIT_RIGHT_FORK_6</text>
    </name>
   </place>
   <place id="cId152">
    <name>
     <text>EAT_2</text>
    </name>
   </place>
   <place id="cId167">
    <name>
     <text>THINK_6</text>
    </name>
    <initialMarking>
     <text>1</text>
    </initialMarking>
   </place>
   <place id="cId163">
    <name>
     <text>WAIT_RIGHT_FORK_1</text>
    </name>
   </place>
   <place id="cId160">
    <name>
     <text>FORK_6</text>
    </name>
    <initialMarking>
     <text>1</text>
    </initialMarking>
   </place>
   <place id="cId174">
    <name>
     <text>WAIT_LEFT_FORK_4</text>
    </name>
   </place>
   <place id="cId149">
    <name>
     <text>EAT_3</text>
    </name>
   </place>
   <place id="cId178">
    <name>
     <text>EAT_1</text>
    </name>
   </place>
   <place id="cId168">
    <name>
     <text>WAIT_RIGHT_FORK_3</text>
    </name>
   </place>
   <place id="cId159">
    <name>
     <text>THINK_2</text>
    </name>
    <initialMarking>
     <text>1</text>
    </initialMarking>
   </place>
   <place id="cId156">
    <name>
     <text>FORK_4</text>
    </name>
    <initialMarking>
     <text>1</text>
    </initialMarking>
   </place>
   <place id="cId158">
    <name>
     <text>WAIT_LEFT_FORK_5</text>
    </name>
   </place>
   <place id="cId150">
    <name>
     <text>FORK_2</text>
    </name>
    <initialMarking>
     <text>1</text>
    </initialMarking>
   </place>
   <place id="cId151">
    <name>
     <text>THINK_3</text>
    </name>
    <initialMarking>
     <text>1</text>
    </initialMarking>
   </place>
   <place id="cId155">
    <name>
     <text>WAIT_LEFT_FORK_6</text>
    </name>
   </place>
   <place id="cId153">
```

```
 <name>
  <text>WAIT_RIGHT_FORK_4</text>
 </name>
</place>
<place id="cId173">
 <name>
  <text>WAIT_LEFT_FORK_2</text>
 </name>
</place>
<place id="cId164">
 <name>
  <text>FORK_5</text>
 </name>
 <initialMarking>
  <text>1</text>
 </initialMarking>
</place>
<place id="cId170">
 <name>
  <text>EAT_6</text>
 </name>
</place>
<place id="cId176">
 <name>
  <text>THINK_5</text>
 </name>
 <initialMarking>
  <text>1</text>
 </initialMarking>
</place>
<place id="cId154">
 <name>
  <text>WAIT_LEFT_FORK_1</text>
 </name>
</place>
<place id="cId157">
 <name>
  <text>THINK_1</text>
 </name>
 <initialMarking>
  <text>1</text>
 </initialMarking>
</place>
<transition id="cId183">
 <name>
  <text>TAKE_LEFT_1_FORK_6</text>
 </name>
</transition>
<transition id="cId180">
 <name>
  <text>RELEASE_FORK_2</text>
 </name>
</transition>
<transition id="cId206">
 <name>
  <text>TAKE_LEFT_1_FORK_4</text>
 </name>
</transition>
<transition id="cId197">
 <name>
  <text>TAKE_RIGHT_2_FORK_2</text>
 </name>
</transition>
<transition id="cId185">
 <name>
  <text>RELEASE_FORK_3</text>
 </name>
</transition>
<transition id="cId196">
 <name>
  <text>TAKE_LEFT_1_FORK_5</text>
 </name>
</transition>
<transition id="cId190">
 <name>
  <text>TAKE_LEFT_1_FORK_2</text>
 </name>
</transition>
<transition id="cId201">
 <name>
  <text>TAKE_LEFT_1_FORK_3</text>
 </name>
</transition>
<transition id="cId187">
 <name>
  <text>TAKE_RIGHT_1_FORK_2</text>
 </name>
</transition>
<transition id="cId194">
 <name>
  <text>TAKE_LEFT_2_FORK_4</text>
 </name>
</transition>
<transition id="cId191">
 <name>
  <text>TAKE_LEFT_2_FORK_3</text>
 </name>
</transition>
<transition id="cId204">
 <name>
  <text>RELEASE_FORK_4</text>
 </name>
</transition>
<transition id="cId202">
 <name>
  <text>TAKE_LEFT_1_FORK_1</text>
 </name>
</transition>
<transition id="cId188">
 <name>
  <text>RELEASE_FORK_6</text>
 </name>
</transition>
<transition id="cId184">
 <name>
  <text>TAKE_LEFT_2_FORK_1</text>
 </name>
</transition>
<transition id="cId208">
 <name>
  <text>TAKE_RIGHT_2_FORK_6</text>
 </name>
</transition>
<transition id="cId205">
 <name>
  <text>TAKE_LEFT_2_FORK_6</text>
 </name>
</transition>
<transition id="cId192">
 <name>
  <text>RELEASE_FORK_5</text>
 </name>
</transition>
<transition id="cId181">
 <name>
  <text>TAKE_RIGHT_2_FORK_5</text>
 </name>
</transition>
<transition id="cId200">
 <name>
  <text>RELEASE_FORK_1</text>
 </name>
</transition>
<transition id="cId193">
 <name>
  <text>TAKE_RIGHT_1_FORK_6</text>
 </name>
</transition>
<transition id="cId207">
 <name>
```

```
  <text>TAKE_RIGHT_2_FORK_3</text>
 </name>
</transition>
<transition id="cId198">
 <name>
  <text>TAKE_LEFT_2_FORK_5</text>
 </name>
</transition>
<transition id="cId182">
 <name>
  <text>TAKE_LEFT_2_FORK_2</text>
 </name>
</transition>
<transition id="cId199">
 <name>
  <text>TAKE_RIGHT_1_FORK_1</text>
 </name>
</transition>
<transition id="cId179">
 <name>
  <text>TAKE_RIGHT_1_FORK_4</text>
 </name>
</transition>
<transition id="cId203">
 <name>
  <text>TAKE_RIGHT_1_FORK_5</text>
 </name>
</transition>
<transition id="cId189">
 <name>
  <text>TAKE_RIGHT_2_FORK_1</text>
 </name>
</transition>
<transition id="cId186">
 <name>
  <text>TAKE_RIGHT_1_FORK_3</text>
 </name>
</transition>
<transition id="cId195">
 <name>
  <text>TAKE_RIGHT_2_FORK_4</text>
 </name>
</transition>
<arc id="cId260" source="cId172" target="cId181"/>
<arc id="cId296" source="cId175" target="cId202"/>
<arc id="cId247" source="cId202" target="cId163"/>
<arc id="cId212" source="cId154" target="cId184"/>
<arc id="cId243" source="cId183" target="cId161"/>
<arc id="cId269" source="cId198" target="cId165"/>
<arc id="cId240" source="cId160" target="cId189"/>
<arc id="cId237" source="cId205" target="cId170"/>
<arc id="cId274" source="cId150" target="cId182"/>
<arc id="cId235" source="cId176" target="cId196"/>
<arc id="cId224" source="cId204" target="cId162"/>
<arc id="cId251" source="cId151" target="cId201"/>
<arc id="cId228" source="cId173" target="cId182"/>
<arc id="cId276" source="cId195" target="cId177"/>
<arc id="cId271" source="cId178" target="cId200"/>
<arc id="cId239" source="cId191" target="cId149"/>
<arc id="cId252" source="cId189" target="cId178"/>
<arc id="cId225" source="cId200" target="cId160"/>
<arc id="cId280" source="cId196" target="cId172"/>
<arc id="cId245" source="cId164" target="cId196"/>
<arc id="cId241" source="cId192" target="cId164"/>
<arc id="cId222" source="cId188" target="cId167"/>
<arc id="cId275" source="cId156" target="cId203"/>
<arc id="cId266" source="cId164" target="cId193"/>
<arc id="cId215" source="cId180" target="cId159"/>
<arc id="cId258" source="cId199" target="cId154"/>
<arc id="cId292" source="cId184" target="cId178"/>
<arc id="cId302" source="cId155" target="cId205"/>
<arc id="cId267" source="cId152" target="cId180"/>
<arc id="cId234" source="cId166" target="cId191"/>
<arc id="cId213" source="cId187" target="cId173"/>
<arc id="cId263" source="cId160" target="cId205"/>
<arc id="cId250" source="cId156" target="cId194"/>
<arc id="cId254" source="cId188" target="cId160"/>
<arc id="cId270" source="cId153" target="cId195"/>
<arc id="cId253" source="cId185" target="cId171"/>
<arc id="cId295" source="cId171" target="cId191"/>
<arc id="cId298" source="cId168" target="cId207"/>
<arc id="cId287" source="cId150" target="cId186"/>
<arc id="cId297" source="cId165" target="cId192"/>
<arc id="cId289" source="cId208" target="cId170"/>
<arc id="cId300" source="cId164" target="cId198"/>
<arc id="cId248" source="cId171" target="cId201"/>
<arc id="cId233" source="cId175" target="cId187"/>
<arc id="cId299" source="cId159" target="cId187"/>
<arc id="cId278" source="cId170" target="cId188"/>
<arc id="cId211" source="cId171" target="cId195"/>
<arc id="cId281" source="cId192" target="cId176"/>
<arc id="cId242" source="cId167" target="cId193"/>
<arc id="cId259" source="cId164" target="cId208"/>
<arc id="cId279" source="cId194" target="cId177"/>
<arc id="cId282" source="cId161" target="cId208"/>
<arc id="cId277" source="cId176" target="cId203"/>
<arc id="cId285" source="cId200" target="cId175"/>
<arc id="cId255" source="cId185" target="cId150"/>
<arc id="cId273" source="cId182" target="cId152"/>
<arc id="cId283" source="cId159" target="cId190"/>
<arc id="cId272" source="cId177" target="cId204"/>
<arc id="cId288" source="cId180" target="cId175"/>
<arc id="cId265" source="cId162" target="cId179"/>
<arc id="cId238" source="cId158" target="cId198"/>
<arc id="cId214" source="cId171" target="cId179"/>
<arc id="cId236" source="cId174" target="cId194"/>
<arc id="cId220" source="cId200" target="cId157"/>
<arc id="cId257" source="cId160" target="cId183"/>
<arc id="cId256" source="cId162" target="cId206"/>
<arc id="cId226" source="cId186" target="cId166"/>
<arc id="cId261" source="cId167" target="cId183"/>
<arc id="cId218" source="cId150" target="cId190"/>
<arc id="cId301" source="cId190" target="cId169"/>
<arc id="cId230" source="cId163" target="cId189"/>
<arc id="cId232" source="cId157" target="cId199"/>
<arc id="cId246" source="cId203" target="cId158"/>
<arc id="cId221" source="cId193" target="cId155"/>
<arc id="cId284" source="cId197" target="cId152"/>
<arc id="cId290" source="cId204" target="cId156"/>
<arc id="cId304" source="cId175" target="cId197"/>
<arc id="cId227" source="cId175" target="cId184"/>
<arc id="cId244" source="cId149" target="cId185"/>
<arc id="cId262" source="cId192" target="cId156"/>
<arc id="cId293" source="cId179" target="cId174"/>
<arc id="cId303" source="cId150" target="cId207"/>
<arc id="cId268" source="cId207" target="cId149"/>
<arc id="cId223" source="cId180" target="cId150"/>
<arc id="cId219" source="cId201" target="cId168"/>
<arc id="cId229" source="cId156" target="cId206"/>
<arc id="cId294" source="cId204" target="cId171"/>
<arc id="cId249" source="cId157" target="cId202"/>
<arc id="cId286" source="cId206" target="cId153"/>
<arc id="cId291" source="cId156" target="cId181"/>
<arc id="cId231" source="cId160" target="cId199"/>
<arc id="cId209" source="cId169" target="cId197"/>
<arc id="cId210" source="cId185" target="cId151"/>
<arc id="cId264" source="cId181" target="cId165"/>
<arc id="cId217" source="cId188" target="cId164"/>
<arc id="cId216" source="cId151" target="cId186"/>
  </page>
 </net>
</pnml>
```

4.3.5 PNML extension for representing subnets

In this section I am going to define new tags and structures in PNML. At this point, I have developed all the necessary to extend PNML in order to represent subnets.

Let's take the following simple Petri net that will serve us to explain the method to achieve a subnet representation and the PNML extension associated to it:

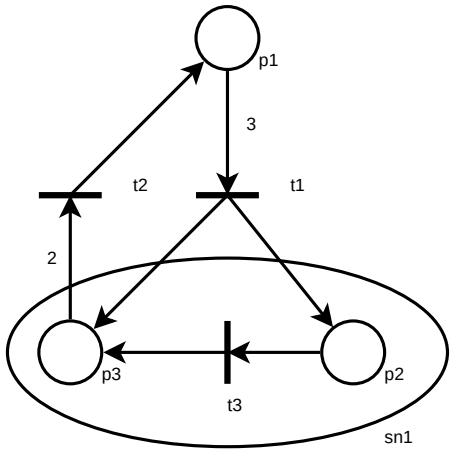


FIGURE 4.2: Subnet to represent in PNML

The PNML code for this net is:

```
<place id="p1"/>
<place id="p2"/>
<place id="p3"/>
<transition id="t1"/>
<transition id="t2"/>
<transition id="t3"/>
<arc id="a1" source="p1" target="t1">
  <inscription>
    <text> 3 </text>
  </inscription>
</arc>
<arc id="a2" source="t1" target="p2"/>
<arc id="a3" source="t1" target="p3"/>
<arc id="a4" source="p3" target="t2">
  <inscription>
    <text>2</text>
  </inscription>
</arc>
<arc id="a5" source="t3" target="p3"/>
<arc id="a6" source="p2" target="t3"/>
<arc id="a7" source="t2" target="p1"/>
```

I want the ellipse region to be a subnet, so I have to specify a subnet with the elements inside the ellipse.

The first step is to define a new tag `<subnet>`. This tag will have an id, as the rest of PNML elements. And now we proceed in this way:

1. The places and transitions inside the subnet are moved into the <subnet> tag
2. The arcs joining subnet elements will be included inside the tag
3. The arcs entering or leaving the subnet will be copied inside the tag. This mean that there are arcs duplicated inside and outside the tag

If we apply these rules to the example:

1. $p2$, $p3$ and $t3$ are moved into the tag `<subnet>`
2. $a5$ and $a6$ are put inside the tag
3. $a2$, $a3$ and $a4$ are copied inside the tag

and we have this other PNML extended code:

```
<subnet id="sn1">
  <place id="p2"/>
  <place id="p3"/>
  <transition id="t3"/>
  <arc id="a2" source="t1" target="p2"/>
  <arc id="a3" source="t1" target="p3"/>
  <arc id="a4" source="p3" target="t2">
    <inscription>
      <text> 2 </text>
    </inscription>
  </arc>
  <arc id="a5" source="t3" target="p3"/>
  <arc id="a6" source="p2" target="t3"/>
</subnet>
<place id="p1"/>
<transition id="t1"/>
<transition id="t2"/>
<arc id="a1" source="p1" target="t1">
  <inscription>
    <text> 3 </text>
  </inscription>
</arc>
<arc id="a2" source="t1" target="p2"/>
<arc id="a3" source="t1" target="p3"/>
<arc id="a4" source="p3" target="t2">
  <inscription>
```

```
    <text> 2 </text>
  </inscription>
</arc>
<arc id="a7" source="t2" target="p1"/>
```

Now this is one of the most important moment in this work: I will separate the inside and the outside of the subnet completely. Taking advantage of the process described in chapter 3, I have to extract the front-end from this subnet.

In this case I have two igp (input gate to a place) and an ogt (output gate to a transition) with weight 2. The figure 4.3 illustrates the interface associated:

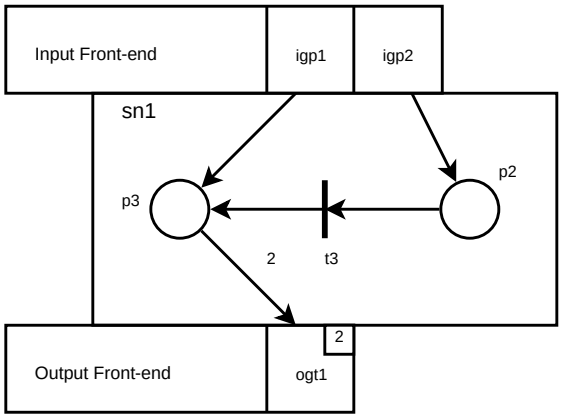


FIGURE 4.3: Subnet with its interface

And the figure 4.4 is the complete net including the subnet and its front-end

And now that I have the graphic, how can I represent it in PNML? To answer this question I will define four new tags: `<interface>`, `<gate>`, <inscription> and `<content>`. Let's explain them.

As its name says, `<interface>` is the tag name for encapsulate the front-end. This tag has no attributes (just the id, of course) but it has embedded the gates inside of it. This gates are represented by `<gate>`. This tag has two new attributes: `action` and `type`. These two attributes have information about the gates. The attribute `action` can take two different values: `input` and `output`. It indicates whether the gate is an input gate or an output gate. The other attribute, `type`, can take other two values: `place` and `transition`. As arcs have weight, gates have it too. For being in accordance, I define the tag `<inscription>` embedded in the tag `<gate>`. It has the weight of the arc associated.

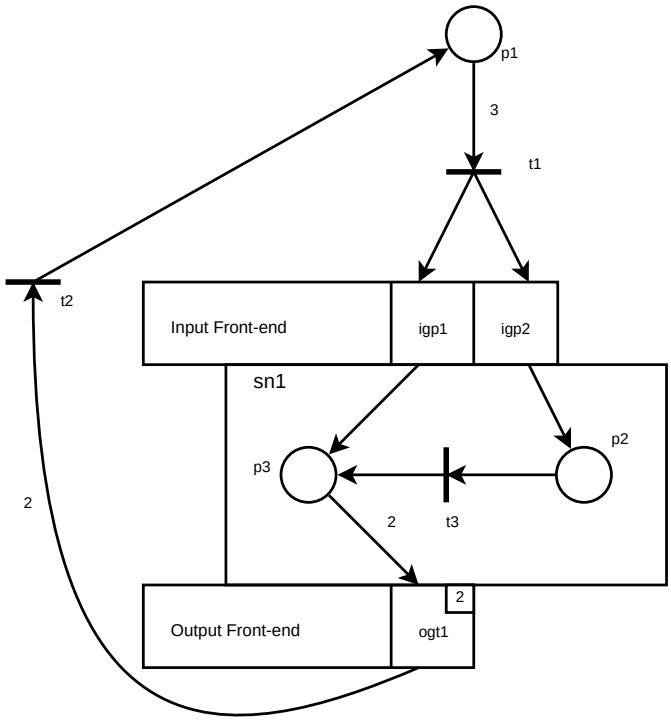


FIGURE 4.4: Petri net with subnet

There is one other tag `<content>` that probably at this moment seems useless, but it is necessary for the rest of the process, as we will see in the next chapter. So I am going to introduce it now. This tag is used to encapsulate the rest of the subnet outside the interface. That is, `<subnet>` has two children: `<interface>` and `<content>`, that have the input/output gates and the rest of the elements, respectively.

Applying this definition to the example, we will have this code

```
<subnet id="sn1">
  <interface id="sn1-interface">
    <gate id="igp1" action="input" type="place"/>
    <gate id="igp2" action="input" type="place"/>
    <gate id="ogt1" action="output" type="transition">
      <inscription>
        <text> 2 </text>
      </inscription>
    </gate>
  </interface>
  <content id="sn1-content">
    <place id="p2"/>
    <place id="p3"/>
    <transition id="t3"/>
    <arc id="a2" source="t1" target="p2"/>
    <arc id="a3" source="t1" target="p3"/>
    <arc id="a4" source="p3" target="t2">
```

```
      <inscription>
        <text> 2 </text>
      </inscription>
    </arc>
    <arc id="a5" source="t3" target="p3"/>
    <arc id="a6" source="p2" target="t3"/>
  </content>
</subnet>
<place id="p1"/>
<transition id="t1"/>
<transition id="t2"/>
<arc id="a1" source="p1" target="t1">
  <inscription>
    <text> 3 </text>
  </inscription>
</arc>
<arc id="a2" source="t1" target="p2"/>
<arc id="a3" source="t1" target="p3"/>
<arc id="a4" source="p3" target="t2">
  <inscription>
    <text> 2 </text>
  </inscription>
</arc>
<arc id="a7" source="t2" target="p1"/>
```

At this moment I have to do only one thing more. The last step is to modify the arcs that are repeated inside and outside the net changing their source or target, depending on where is it:

- If the arc is **entering** the subnet
  - For the `<arc>` tag inside the tag `<subnet>`, the **source** attribute of the arc is changed by the **input** gate associated
  - For the `<arc>` tag outside the tag `<subnet>`, the **target** attribute of the arc is changed by the **output** gate associated
- If the arc is **leaving** the subnet
  - For the `<arc>` tag inside the tag `<subnet>`, the **target** attribute of the arc is changed by the **input** gate associated
  - For the `<arc>` tag outside the tag `<subnet>`, the **source** attribute of the arc is changed by the **output** gate associated

Applying again this rule to the example we have the definitive code for this Petri net:

```
<subnet id="sn1">
  <interface id="sn1-interface">
    <gate id="igp1" action="input" type="place"/>
```

```
    <gate id="igp2" action="input" type="place"/>
    <gate id="ogt1" action="output" type="transition">
      <inscription>
        <text> 2 </text>
      </inscription>
    </gate>
  </interface>
  <content id="sn1-content">
    <place id="p2"/>
    <place id="p3"/>
    <transition id="t3"/>
    <arc id="a2" source="igp2" target="p2"/>
    <arc id="a3" source="igp1" target="p3"/>
    <arc id="a4" source="p3" target="ogt1">
      <inscription>
        <text> 2 </text>
      </inscription>
    </arc>
    <arc id="a5" source="t3" target="p3"/>
    <arc id="a6" source="p2" target="t3"/>
  </content>
</subnet>
<place id="p1"/>
<transition id="t1"/>
<transition id="t2"/>
<arc id="a1" source="p1" target="t1">
  <inscription>
    <text> 3 </text>
  </inscription>
</arc>
<arc id="a2" source="t1" target="igp2"/>
<arc id="a3" source="t1" target="igp1"/>
<arc id="a4" source="ogt1" target="t2">
  <inscription>
    <text> 2 </text>
  </inscription>
</arc>
<arc id="a7" source="t2" target="p1"/>
```

Listing 4.8: Final PNML representation

Once this is done, the only way to enter or leave the subnet is crossing the front-end, and I have reached my goal.

This is a comprehensive definition of how to represent subnets in PNML. Now there are several ways to create a grammar extension that frame this structure of xml. For example, we can define a dtd file[77], and xsd file [78, 79] or, by coherence with the original grammar of PNML, a Relax NG file [80]. I have defined a way to represent subnets, but the formal grammar is outside the scope of my work because of the wide casuistry of these Petri net types. However, the method is explained enough on order to each one of these types to define their ox extension.

//TODO gramatica de las subredes para RdP basicas?? Si da tiempo

4.3.5.1 Examples of PNML subnets

//TODO

4.4 Conclusions

In this section we have seen a way to represent subnets in PNML format. It has not been a formal definition, but general guidelines in order to make a more extended study of the Petri nets. There are lots of different Petri net types. Each one of them has their own particularities that have to be translated into the PNML format. The method explained here is based on the basic general Petri nets, but it can see viewed as an algorithm that allow these types to define their own tags.

Basically, for new xml tags are introduced

- `<subnet>` for delimitate the scope (places and transitions) of the subnet
- <interface> for define the gates to enter or leave the subnet. Each arc entering or leaving the subnet is associated to a specific gate. Each gate is represented with the tag `<gate>` and can be of several types: input/output (depending on if the arcs enters or leaves the subnet) and place/transition (depending on if the arc is associated to a place or a transition inside the subnet)
- `<content>` where the places and transitions inside the subnet are placed, in addition to the arcs between them.

The really important thing in this chapter is to identify the subnet and process it in order to extract and interface that is the only way to enter and leave the subnet. The details of each one of the different Petri net types are not specified because of the big casuistry of them.

Once the method is applied, there has to be one subnet content and one subnet interface. There two rules that must be accomplished:

1. no arc can join nodes inside the subnet with nodes outside and viceversa
2. the interface has to support all the arcs entering and leaving the subnet with the same information of the arcs that it replace.

If these two rules are complied, then the subnet is correctly defined. Obviously, this method can be applied several times in order to declare several subnets. the only restriction is that each place or transition can be only in one subnet: no place/transition can be stored in two or more

Chapter 5

Security

5.1 Introduction

This is the second main goal, after subnetting. Once the possible subnets are defined it is the turn of securing them. It is possible to secure Petri subnets or the entire net.

With secure, I mean four goals:

- **Privacy**. Concrete parts of the net must be occulted: The content is secret, so not everybody should be able to know it.
- **Integrity**. Any change in the secured parts has to be detected. If it suffers any kind of modification, the information may have been compromised, and perhaps it is not valid or correct. But I cannot know what has been modified: I can only detect that the original content has been changed.
- **Authentication**. I can authenticate the source of that (the signer, author or guarantor).
- **Non repudiation**. With this characteristic, avoiding the possibility of supplant other people is avoided. So the person that sign that part cannot say that he hadn't done it.The signer cannot deny it.

There can be several reasons for hiding information of a Petri net. For example:

- One subnet is a secret process that I want to hide from indiscrete eyes
- I have a main process that communicates with other processes. These processes are susceptible to be changed and the only information I need is the interface, so they can be easily replaced by other implementations.

But this information should be accessible to authorized people without necessity of supplying any other kind of data. So the whole information may be stored in the same file.

In the same way, there can be many reasons for the rest of the security characteristics. For example, suppose that we have a Petri net that several people can access an modify:

- Some parts of that Petri net have been validated an accepted, so I want nobody to change them. In this case **integrity** is needed.
- I want to know who has developed a concrete chunk of the net. **Authentication** is required.
- There is a part of the Petri net that is bad defined and go wrong. The person responsible of it says that he hasn't made it and somebody has supplant him. Then, **non repudiation** is needed.

The best way to reach these goals is using standard and proved technologies. In this case, the selected technologies are:

- XMLEncryption[75] for privacy
- XMLSignature [76] for integrity, authentication and non repudiation

5.2 XMLEncryption

5.2.1 XMLEncryption revision

XMLEncryption is a World Wide Web Consortium (W3C) Recommendation form encrypt xml or non xml content. It is an xml file ciphering standard. Both symmetric and asymmetric ciphering can be used, but in this case, symmetric is preferred. The main idea of this encryption is to replace the xml element or elements we want to be ciphered by other xml code that contains the ciphered data, in addition to information of the algorithms used for the encryption process. When a non xml file in ciphered, the only option is to encrypt it completely. But, when it is applied to xml content, this technology allows us to define concrete fragments of the document we want to hide. Moreover, the xml document can be transformed before applying the encryption, for example, in order to format the normalize the xml content. In this work, the pieces of xml content susceptible to be ciphered are, obviously, the subnets represented in PNML format.

Regardless of the data source (xml or non xml) the result is always a xml element. Normal is that this xml encrypted chunk have the whole necessary information to be decrypted. Among that information we can find:

- Ciphering algorithm: it is the name of chosen method for ciphering the data. It can be not included. In this case, both ciphering and deciphering agents have to know which is the exact this algorithm.
- The ciphered data: obviously this parts is mandatory and has always to be present.
- Name of the chosen key: it is optional. It is used when a set of keys is known by both ciphering an deciphering agents.
- Key: it is optional. In this case there is a symmetric key in order to encrypt the data and an additional pair of keys: one (known by the cipher agent) to encrypt the symmetric key and the other (known by the decipher agent) to decrypt it.

Actually, there are several options to apply XMLEncryption, such as the algorithm or the key. The exact election of those option values is responsibility of the Petri net sender. For example:

- Maybe both parts (sender and receiver) have a common set of keys, so they can use it in order to encrypt and decrypt the subnet.
- Other common use is that the key is defined inside the options but it is ciphered itself. If the receiver have a pair of keys (public and private) and an asymmetrical algorithm (such as Diffie-Hellman or RSA) , the symmetric key can be ciphered by the sender with the receiver's public key. In this case, only the receiver can decrypt it using his private key.

This section does not want to be an extensive explanation about XMLEncryption but a general idea about its functionality. So I am not going to deepen the whole characteristics of XMLEncryption. The final decision about which options use is responsibility of those people that want to apply this work, basing their decision on the requirements of their own Petri net.

Once this is said, here we have a basic example of XMLEncryption. Let's take this original xml document:

```
<?xml version='1.0'?>
<PaymentInfo xmlns='http://example.org/paymentv2'>
  <Name>John Smith</Name>
  <CreditCard Limit='5,000' Currency='USD'>
```

```
    <Number>4019 2445 0277 5567</Number>
    <Issuer>Example Bank</Issuer>
    <Expiration>04/02</Expiration>
  </CreditCard>
</PaymentInfo>
```

LISTING 5.1: Clear xml content

In this example, we want to hide the credit card number <Number>. Several options are going to be applied: with and without the ciphering information

First of all let's see which will be the aspect of the ciphered content without information about ciphering, only replacing the clear data by the encrypted data: we have not information about the key or the ciphering algorithm. This is the xml ciphered code:

```
<?xml version='1.0'?>
<PaymentInfo xmlns='http://example.org/paymentv2'>
  <Name>John Smith</Name>
  <CreditCard Limit='5,000' Currency='USD'>
    <Number>
      <xenc:EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
          Type='http://www.w3.org/2001/04/xmlenc#Content'>
        <xenc:CipherData>
          <xenc:CipherValue>A23B45C56</CipherValue>
        </CipherData>
      </EncryptedData>
    </Number>
    <Issuer>Example Bank</Issuer>
    <Expiration>04/02</Expiration>
  </CreditCard>
</PaymentInfo>
```

LISTING 5.2: Ciphered xml content without ciphering information

As we can see, the credit card number has been replaced by a new tag <EncryptedData> that contains the ciphered credit card number.

Note. A new namespace **xenc** appear that is the standard namespace for XMLEncryption. However, in futher examples this namespace can be delete for clarity and for space problems without loss of generality.

And now let's see how does this same example with information about the algorithm:

```
<?xml version='1.0'?>
<PaymentInfo xmlns='http://example.org/paymentv2'>
  <Name>John Smith</Name>
  <CreditCard Limit='5,000' Currency='USD'>
    <Number>
      <xenc:EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
          Type='http://www.w3.org/2001/04/xmlenc#Content'>
        <xenc:EncryptionMethod
            Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"
```

```
            xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" />
          <xenc:CipherData>
            <xenc:CipherValue>A23B45C56</CipherValue>
          </CipherData>
        </EncryptedData>
      </Number>
      <Issuer>Example Bank</Issuer>
      <Expiration>04/02</Expiration>
    </CreditCard>
</PaymentInfo>
```

LISTING 5.3: Ciphered xml content with algorithm information

As we can see, a new tag <`EncryptedMethod`> has appeared inside the <`EncryptedData`> tag with the algorithm used to cipher. In this case is `aes128-cbc`.

There is other method of Encryption that cipher the tag too. In this case, we would have the next code:

```
<?xml version='1.0'?>
<PaymentInfo xmlns='http://example.org/paymentv2'>
  <Name>John Smith</Name>
  <CreditCard Limit='5,000' Currency='USD'>
    <xenc:EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
        Type='http://www.w3.org/2001/04/xmlenc#Element'>
      <xenc:EncryptionMethod
          Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"
          xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" />
      <xenc:CipherData>
        <xenc:CipherValue>A223B3B493G5C569M</CipherValue>
      </CipherData>
    </EncryptedData>
    <Issuer>Example Bank</Issuer>
    <Expiration>04/02</Expiration>
  </CreditCard>
</PaymentInfo>
```

LISTING 5.4: Ciphered xml content including the tag itself

As we can see, the tag <`Number`> has disappeared and it has been included into the <`CipherValue`> of the <`CipherData`>.

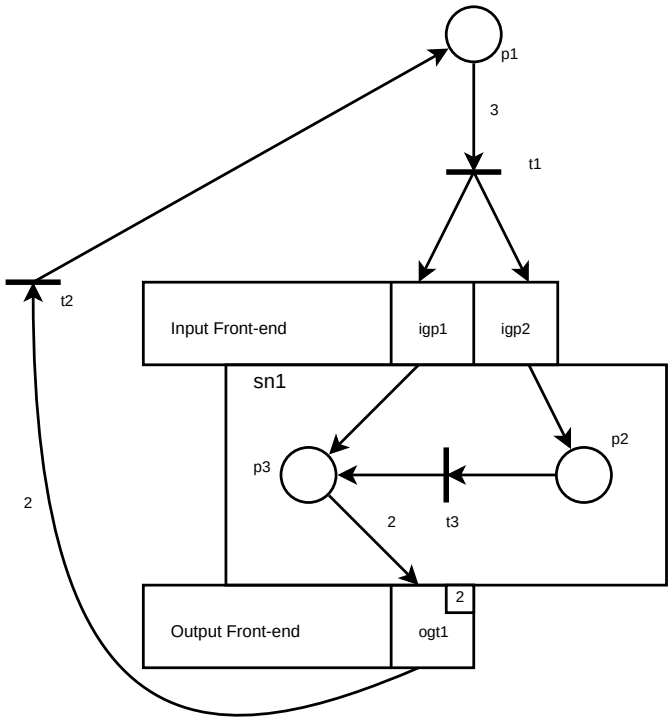
This is a little approach to XMLEncryption functionality, but enough for understanding the next section.

5.2.2 XMLEncryption and Petri nets

Once described XMLEncryption it is time to apply it in order to hide part of a Petri net. Remembering the chapter 4, we have one Petri net with or more subnets represented in a PNML file. These subnets are represented by a <`subnet`> tag that contains

<interface> and <content>. This last tag the xml content that is going to be ciphered. Obviously, if we encrypt the interface we will have no way to connect the subnet with the rest of the net.

Let's take back the example used to explain the process of subnetting in the figure 4.4...



...and its PNML representation

```
<subnet id="sn1">
  <interface id="sn1-interface">
    <gate id="igp1" action="input" type="place"/>
    <gate id="igp2" action="input" type="place"/>
    <gate id="ogt1" action="output" type="transition">
      <inscription>
        <text> 2 </text>
      </inscription>
    </gate>
  </interface>
  <content id="sn1-content">
    <place id="p2"/>
    <place id="p3"/>
    <transition id="t3"/>
    <arc id="a2" source="igp2" target="p2"/>
    <arc id="a3" source="igp1" target="p3"/>
    <arc id="a4" source="p3" target="ogt1">
      <inscription>
        <text> 2 </text>
      </inscription>
    </arc>
    <arc id="a5" source="t3" target="p3"/>
```

```
    <arc id="a6" source="p2" target="t3"/>
  </content>
</subnet>
<place id="p1"/>
<transition id="t1"/>
<transition id="t2"/>
<arc id="a1" source="p1" target="t1">
  <inscription>
    <text> 3 </text>
  </inscription>
</arc>
<arc id="a2" source="t1" target="igp2"/>
<arc id="a3" source="t1" target="igp1"/>
<arc id="a4" source="ogt1" target="t2">
  <inscription>
    <text> 2 </text>
  </inscription>
</arc>
<arc id="a7" source="t2" target="p1"/>
```

The goal is to hide the internal content of the subnet. If we apply XMLEncryption to the data contained inside the tag <`content`>, we will get something like this, depending on the algorithm and key selected for the ciphering.

```
<subnet id="sn1">
  <interface id="sn1-interface">
    <gate id="igp1" action="input" type="place"/>
    <gate id="igp2" action="input" type="place"/>
    <gate id="ogt1" action="output" type="transition">
      <inscription>
        <text> 2 </text>
      </inscription>
    </gate>
  </interface>
  <content id="sn1-content">
    <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
        Type="http://www.w3.org/2001/04/xmlenc#Element">
        <xenc:EncryptionMethod
            Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"
            xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" />
        <xenc:CipherData
            xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
            <xenc:CipherValue
                xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
                Wr1njyJlYYOM91AYqcwGCWkw2L4pUjQD2GGVoU91VZ0wKqHY8y3lGY8FY4i5K
                3GY8FY4i5K3G8grIe1HRFqe7RtkFiXZgGMeYnQp6oB6ckKp3KFKHVqtucc9rA
                VzOgC7XAwe61HRFqe6RRVzXjNM9hlVZ0wKqHY8y3l3GY8FY4i5K3G8grIe2xN
                4u7x7fRtkFiXZgGMeYnQp6oB6ckKp3KFRRVzXjNAtVzOgC7XAw/oe61HRFqe6
                RRVzXjNMLU5ZgGMeYny8NVPQmUSDX7NRtnR6YnQp6oB6GY8F=
            </xenc:CipherValue>
        </xenc:CipherData>
    </xenc:EncryptedData>
  </content>
</subnet>
```

```
<place id="p1"/>
<transition id="t1"/>
<transition id="t2"/>
<arc id="a1" source="p1" target="t1">
  <inscription>
    <text> 3 </text>
  </inscription>
</arc>
<arc id="a2" source="t1" target="igp2"/>
<arc id="a3" source="t1" target="igp1"/>
<arc id="a4" source="ogt1" target="t2">
  <inscription>
    <text> 2 </text>
  </inscription>
</arc>
<arc id="a7" source="t2" target="p1"/>
```

If we try to represent this Petri net we will have the interface of the subnet, but the content is a black box. So it is as follows in the figure:

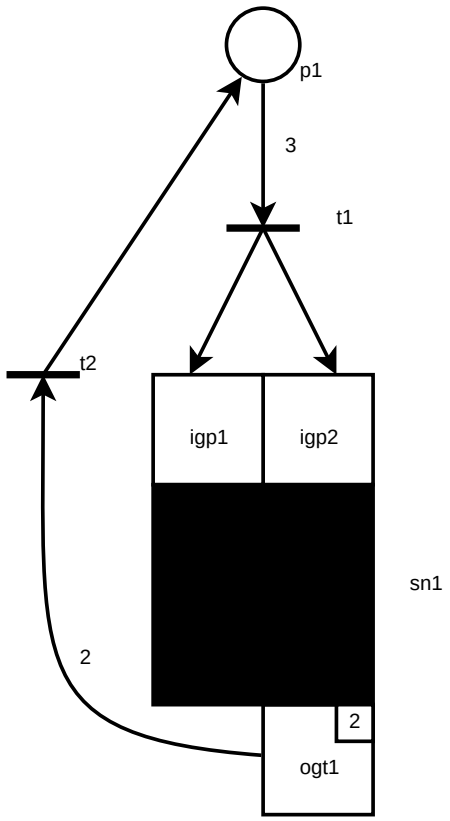


FIGURE 5.1: Petri net with hidden subnet

5.2.3 Examples

5.2.3.1 Hiding several subnets

One important thing is that I can cipher several subnets of the same Petri net with distinct options. For example, if there are two subnets for both two distinct receivers each one of the subnets can be configured in order to each subnet can be decrypted by its own receiver.

//TODO

5.2.3.2 Critical processes

//TODO

5.3 XMLSignature

5.3.1 Introduction

Although privacy is a solved question with XMLEncryption, there are other aspects of the security that XMLEncryption can't cover. For example:

- We want nobody to modify parts of a Petri net
- I want to put on record that I am the author or the responsive of a Petri net
- Maybe I want to be sure that the author of a Petri net is somebody with no doubt

To solve these questions, we can use digital signature. It gives us several interesting characteristics:

- **Integrity**: The data integrity will appear if we are able to avoid, or at least detect, any non authorized modification of the information
- **Authentication**: authentication guarantee that somebody is who says that he is. Basically, with digital signature nobody can impersonate another.
- **Non repudiation**: with this characteristic, we can avoid that somebody says he hasn't signed something if he really did. Only that person can have done it.

In this case I want to sign a whole Petri net or concrete parts of it. Obviously, if I sign only of a Petri net, this part keeps integrity, authentication and non repudiation, but the rest of it doesn't.

As with XMLEncryption, the best way to achieve this goals is using standard technologies. For signing, the method chosen is XMLSignature [76].

5.3.2 XMLSignature revision

XMLSignature is a digital signing standard. With XMLSignature we can sign any kind of content but the result is XML content. It requires the use of digital certificates and a set of public/private keys, using asymmetrical ciphering algorithms for the process.

It has three possibilities:

1. **Enveloped**: The result of the signing is the original xml file with a new attached signature element inside.
2. **Enveloping**: The result of the signing is a new xml file with the digital signature, and, inside it, the original signed elements of the initial xml file.
3. **Detached**: The result of the signature is a new independent file. The original xml file remains inalterable, The new file contains the sign.

It is indifferent which of this three methods use. They are only different ways to organize the generated signature.

XMLSignature forces a digital signature to have:

- **Canonicalization method**: Basically, we have an equivalent relationship that is able to detect if two xml files are equivalent each other. A canonicalization method transforms a xml file into a canonicalized one that is the representant of the equivalence class. All the equivalent xml files are transformed in this representant. This transform has to be applied before signing. If no method is declared, there is one defined by default.
- **Reference**: there can be several references in one only signature. Each reference indicates a part of the document that has to be signed and the digest algorithm used. [1]. It is important to say that each reference doesn't generate a signature, but all of these references are signed together and generates only one signature.

[1] A digest algorithm generates a fixed length bytes sequence from arbitrary length contents. This bytes sequence is different for each content

- **Key information**: Optionally, the signature can include necessary information to be validated. In this part we can indicate the public key directly, by a bytes sequence, by an URL or the most usual X.509 digital certificates.
- **Transforms**: It is possible that that we want to sign is not the whole document, but concrete information obtained from it (but not the content itself): e.g. encoding/decoding, select only certain parts, modify the XML structure, include fragments of other documents,... Transforms is a ordered list of processing steps that has to be applied to the content before being digested. This feature is optional but in this case it will be necessary.

A XML signature consists of a `<Signature>` tag defined in the `http://www.w3.org/2000/09/xmldsig#` namespace.

The basic structure of the element signature is:

```
<Signature>
  <SignedInfo>
    <CanonicalizationMethod />
    <SignatureMethod />
    <Reference>
      <Transforms>
        <Transform />
      </Transforms>
      <DigestMethod>
      <DigestValue>
    </Reference>
    <Reference /> etc.
  </SignedInfo>
  <SignatureValue />
  <KeyInfo />
  <Object />
</Signature>
```

The element `<Object>` is used only in enveloping signature to store the signed data. In this case I am going to use enveloped signature, so this element will no be present in the examples.

The final aspect of a complete XMLSignature element is like this example:

```
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
    <CanonicalizationMethod
        Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"
        xmlns="http://www.w3.org/2000/09/xmldsig#" />
    <SignatureMethod
        Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"
        xmlns="http://www.w3.org/2000/09/xmldsig#" />
    <Reference URI=""
        xmlns="http://www.w3.org/2000/09/xmldsig#">
```

```
      <Transforms
          xmlns="http://www.w3.org/2000/09/xmldsig#">
        <Transform
            Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"
            xmlns="http://www.w3.org/2000/09/xmldsig#" />
      </Transforms>
      <DigestMethod
          Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"
          xmlns="http://www.w3.org/2000/09/xmldsig#" />
      <DigestValue
          xmlns="http://www.w3.org/2000/09/xmldsig#">
        Oyyx+K28+cp7kuUgcnANtTBdUwg=
      </DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue xmlns="http://www.w3.org/2000/09/xmldsig#">
    ZVzRud7G4mEZsDnBavbnZoFUmm5J2OBDkQ+IooDLn95ndGYdrq6uPQ==
  </SignatureValue>
  <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
    <X509Data xmlns="http://www.w3.org/2000/09/xmldsig#">
      <X509Certificate
          xmlns="http://www.w3.org/2000/09/xmldsig#">
        IICmDCCAlYCBEfrim8wCwYHKoZIzjgEAwUAMDIxCzAJBgNVBAYTAkVTMREwDwYDVQQKEwhBd
        pYTEQMA4GA1UEAxMHVXN1YXJpbzAeFw0wODAzMjcxMTUyMTVaFw0wOTAzMjcxMTUyMTVaMDI
        BgNVBAYTAkVTMREwDwYDVQQKEwhBdXRlbnRpYTEQMA4GA1UEAxMHVXN1YXJpbzCCAbcwggEs
        BgcqhkjOOAQBMIIBHwKBgQD9f1OBHXUSKVLfSpwu7OTn9hG3UjzvRADDHj+AtlEmaUVdQCJR
        jVj6v8X1ujD2y5tVbNeBO4AdNG/yZmC3a5lQpaSfn+gEexAiwk+7qdf+t8Yb+DtX58aophUP
        9tPFHsMCNVQTWhaRMvZ1864rYdcq7/IiAxmd0UgBxwIVAJdgUI8VIwvMspK5gqLrhAvwWBz1
        APfhoIXWmz3ey7yrXDa4V7l5lK+7+jrqgvlXTAs9B4JnUVlXjrrUWU/mcQcQgYC0SRZxI+hM
        t88JMozIpuE8FnqLVHyNKOCjrh4rs6Z1kW6jfwv6ITVi8ftiegEkO8yk8b6oUZCJqIPf4Vrl
        i2ZegHtVJWQBTDv+z0kqA4GEAAKBgDUPDwxDZFXMrZha74VNmgyFslLM01wKw17nbt9UFTJA
        iPpozeZMP2u0SoYst2nbxkCs1hziiuaNJnykzcjVf3+PmL3sQES8SxwJBRUME2UTA2006WD3
        iZ9yibcWQimB8eKIJyBBxSk5TueAzvTA8HN2+Rvgh8RMaOzhMAsGByqGSM44BAMFAAMvADAs
        4+nQZdFvlvsfyOfq1t02h9MJEgIUEvYDfxeygKCmrIlA0sQLtaCs0Qo=
      </X509Certificate>
    </X509Data>
    <KeyValue xmlns="http://www.w3.org/2000/09/xmldsig#">
      <DSAKeyValue
          xmlns="http://www.w3.org/2000/09/xmldsig#">
        <P xmlns="http://www.w3.org/2000/09/xmldsig#">
          /X9TgR11EilS30qcLuzk5/YRt1I870QAwx4/gLZRJmlFXUAiUftZPY1Y+r/F9bow9subVWz
          HTRv8mZgt2uZUKWkn5/oBHsQIsJPu6nX/rfGG/g7V+fGqKYVDwT7g/bTxR7DAjVUE1oWkTL
          K2HXKu/yIgMZndFIAcc=
        </P>
        <Q xmlns="http://www.w3.org/2000/09/xmldsig#">
          l2BQjxUjC8yykrmCouuEC/BYHPU=
        </Q>
        <G xmlns="http://www.w3.org/2000/09/xmldsig#">
          9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT+ZxBxCBgLRJFnEj6Ew
          zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx+2J6ASQ7zKTxvqhRkImog9/hWuW
          Zl6Ae1UlZAFMO/7PSSo=
        </G>
        <Y xmlns="http://www.w3.org/2000/09/xmldsig#">
          NQ8PDENkVcytmFrvhU2aDIWyUszTXArDXudu31QVMkAuTvWI+mjN5kw/a7RKhiy3advGQKz
          5o0mfKTNyNV/f4+YvexARLxLHAkFFQwTZRMDbTTpYPfE3L2Jn3KJtxZCKYHx4ognIEHFKTl
```

```
          9MDwc3b5G+CHxExo7OE=
        </Y>
      </DSAKeyValue>
    </KeyValue>
  </KeyInfo>
</Signature>
```

At first sight it seems very complicated, but it isn't. First of all, the attribute xmlns that is in almost all the tags is only the namespace. From here, for more clarity it can be obviated. So the example is as follows:

```
<Signature>
  <SignedInfo>
    <CanonicalizationMethod
        Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
    <SignatureMethod
        Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1" />
    <Reference URI="">
      <Transforms>
        <Transform
            Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
      </Transforms>
      <DigestMethod
          Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <DigestValue>
        Oyyx+K28+cp7kuUgcnANtTBdUwg=
      </DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>
    ZVzRud7G4mEZsDnBavbnZoFUmm5J2OBDkQ+IooDLn95ndGYdrq6uPQ==
  </SignatureValue>
  <KeyInfo>
    <X509Data>
      <X509Certificate>
        IICmDCCAlYCBEfrim8wCwYHKoZIzjgEAwUAMDIxCzAJBgNVBAYTAkVTMREwDwYDVQQKEwhBd
        pYTEQMA4GA1UEAxMHVXN1YXJpbzAeFw0wODAzMjcxMTUyMTVaFw0wOTAzMjcxMTUyMTVaMDI
        BgNVBAYTAkVTMREwDwYDVQQKEwhBdXRlbnRpYTEQMA4GA1UEAxMHVXN1YXJpbzCCAbcwggEs
        BgcqhkjOOAQBMIIBHwKBgQD9f1OBHXUSKVLfSpwu7OTn9hG3UjzvRADDHj+AtlEmaUVdQCJR
        jVj6v8X1ujD2y5tVbNeBO4AdNG/yZmC3a5lQpaSfn+gEexAiwk+7qdf+t8Yb+DtX58aophUP
        9tPFHsMCNVQTWhaRMvZ1864rYdcq7/IiAxmd0UgBxwIVAJdgUI8VIwvMspK5gqLrhAvwWBz1
        APfhoIXWmz3ey7yrXDa4V7l5lK+7+jrqgvlXTAs9B4JnUVlXjrrUWU/mcQcQgYC0SRZxI+hM
        t88JMozIpuE8FnqLVHyNKOCjrh4rs6Z1kW6jfwv6ITVi8ftiegEkO8yk8b6oUZCJqIPf4Vrl
        i2ZegHtVJWQBTDv+z0kqA4GEAAKBgDUPDwxDZFXMrZha74VNmgyFslLM01wKw17nbt9UFTJA
        iPpozeZMP2u0SoYst2nbxkCs1hziiuaNJnykzcjVf3+PmL3sQES8SxwJBRUME2UTA2006WD3
        iZ9yibcWQimB8eKIJyBBxSk5TueAzvTA8HN2+Rvgh8RMaOzhMAsGByqGSM44BAMFAAMvADAs
        4+nQZdFvlvsfyOfq1t02h9MJEgIUEvYDfxeygKCmrIlA0sQLtaCs0Qo=
      </X509Certificate>
    </X509Data>
    <KeyValue>
      <DSAKeyValue>
        <P>
          /X9TgR11EilS30qcLuzk5/YRt1I870QAwx4/gLZRJmlFXUAiUftZPY1Y+r/F9bow9subVWz
          HTRv8mZgt2uZUKWkn5/oBHsQIsJPu6nX/rfGG/g7V+fGqKYVDwT7g/bTxR7DAjVUE1oWkTL
```

```
          K2HXKu/yIgMZndFIAcc=
        </P>
        <Q>
          l2BQjxUjC8yykrmCouuEC/BYHPU=
        </Q>
        <G>
          9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT+ZxBxCBgLRJFnEj6Ew
          zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx+2J6ASQ7zKTxvqhRkImog9/hWuW
          Zl6Ae1UlZAFMO/7PSSo=
        </G>
        <Y>
          NQ8PDENkVcytmFrvhU2aDIWyUszTXArDXudu31QVMkAuTvWI+mjN5kw/a7RKhiy3advGQKz
          5o0mfKTNyNV/f4+YvexARLxLHAkFFQwTZRMDbTTpYPfE3L2Jn3KJtxZCKYHx4ognIEHFKTl
          9MDwc3b5G+CHxExo7OE=
        </Y>
      </DSAKeyValue>
    </KeyValue>
  </KeyInfo>
</Signature>
```

5.3.3 XMLSignature and Petri nets

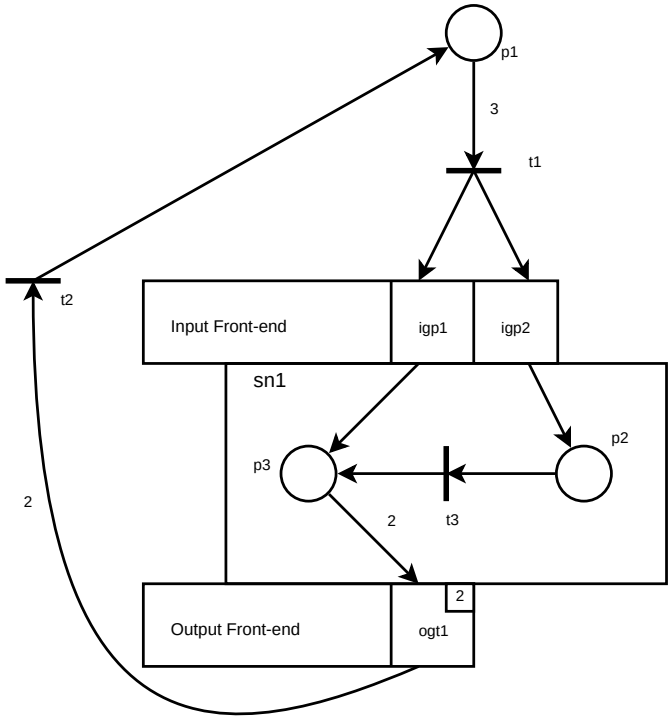
In this case I am going to use enveloped signature, so the result of the signature is stored in a new tag inside the original PNML file. As I explained before, we have to select which parts of the PNML file are going to be signed.

Many times we will need to sign the whole Petri net, but it will be very usual to sign only certain parts of a Petri net, for example a critical subprocess. The modus operandi here is similar to XMLEncryption. First of all, the content to be signed should be grouped in a subnet and then, this subnet is signed.

The standard way to indicate a subnet to sign in XMLSignature is through a XPath [81] expression. In XMLSignature, the way to specify XPath addresses is using XMLSignature XPath Filter [82]. XPathFilter returns the node set that is going to be signed and it is placed into `/Signature/SignedInfo/Reference/Transforms` as a new `<Transform>`.

I am not going to explain all the possibilities of XPath Filter. I will explain only those main configurations useful to my objective. The exact configuration depends on the particular necessities of each case.

In order to illustrate the process, let's take the figure 4.4 as example again:



and its full PNML representation:

```
<?xml version="1.0" encoding="utf-8"?>
<pnml>
  <net id="myNet" type="http://www.pnml.org/version-2009/grammar/ptnet">
    <name>
      <text> My new net </text>
    </name>
    <page id="page1">
      <subnet id="sn1">
        <interface id="sn1-interface">
        <gate id="igp1" action="input" type="place"/>
        <gate id="igp2" action="input" type="place"/>
        <gate id="ogt1" action="output" type="transition">
          <inscription>
          <text> 2 </text>
          </inscription>
        </gate>
        </interface>
        <content id="sn1-content">
        <place id="p2"/>
        <place id="p3"/>
        <transition id="t3"/>
        <arc id="a2" source="igp2" target="p2"/>
        <arc id="a3" source="igp1" target="p3"/>
        <arc id="a4" source="p3" target="ogt1">
          <inscription>
          <text> 2 </text>
          </inscription>
```

```
        </arc>
        <arc id="a5" source="t3" target="p3"/>
        <arc id="a6" source="p2" target="t3"/>
        </content>
      </subnet>
      <place id="p1"/>
      <transition id="t1"/>
      <transition id="t2"/>
      <arc id="a1" source="p1" target="t1">
        <inscription>
        <text> 3 </text>
        </inscription>
      </arc>
      <arc id="a2" source="t1" target="igp2"/>
      <arc id="a3" source="t1" target="igp1"/>
      <arc id="a4" source="ogt1" target="t2">
        <inscription>
        <text> 2 </text>
        </inscription>
      </arc>
      <arc id="a7" source="t2" target="p1"/>
    </page>
  </net>
</pnml>
```

The first option is to sign the whole net. In XPath, the expression to represent the entire document is:

```
/
```

If we apply XMLSignature with this XPath expression, the result is

```
<?xml version="1.0" encoding="UTF-8"?>
<pnml>
  <net id="myNet" type="http://www.pnml.org/version-2009/grammar/ptnet">
    <name>
      <text> My new net </text>
    </name>
    <page id="page1">
      <subnet id="sn1">
        <interface id="sn1-interface">
          <gate action="input" id="igp1" type="place"/>
          <gate action="input" id="igp2" type="place"/>
          <gate action="output" id="ogt1" type="transition">
            <inscription>
              <text> 2 </text>
            </inscription>
          </gate>
        </interface>
        <content id="sn1-content">
          <place id="p2"/>
          <place id="p3"/>
          <transition id="t3"/>
          <arc id="a2" source="igp2" target="p2"/>
          <arc id="a3" source="igp1" target="p3"/>
          <arc id="a4" source="p3" target="ogt1">
            <inscription>
              <text> 2 </text>
            </inscription>
          </arc>
```

```
          <arc id="a5" source="t3" target="p3"/>
          <arc id="a6" source="p2" target="t3"/>
        </content>
      </subnet>
      <place id="p1"/>
      <transition id="t1"/>
      <transition id="t2"/>
      <arc id="a1" source="p1" target="t1">
        <inscription>
          <text> 3 </text>
        </inscription>
      </arc>
      <arc id="a2" source="t1" target="igp2"/>
      <arc id="a3" source="t1" target="igp1"/>
      <arc id="a4" source="ogt1" target="t2">
        <inscription>
          <text> 2 </text>
        </inscription>
      </arc>
      <arc id="a7" source="t2" target="p1"/>
    </page>
  </net>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <ds:Reference URI="">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
          <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
          <ds:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
            <dsig-xpath:XPath xmlns:dsig-xpath="http://www.w3.org/2002/06/xmldsig-filter2" Filter="union">
              /
            </dsig-xpath:XPath>
          </ds:Transform>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>LIMSPHwtGpK3hlDEdfaAv7D39KU=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>
      W3C5nlPSYcR2qvx2bOwpGy8CGMu0vWPTZCIQVkYuGUA52lRmtyFRInlRm0G+d+eqQHBwtvMophte
      HWPsPiOl+Bwc/C3HTHj2XBc9P1bcFUtVM91rLhLZhI/ZAl9t6VfLoQ+Cduu8sQJh6qiH24CiYGjc
      FaOlQbQOsYBvGXoBhEk=
    </ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509Certificate>
          MIICgTCCAeqgAwIBAgIETfh4CTANBgkqhkiG9w0BAQUFADCBhDELMAkGA1UEBhMCRVMxETAPBgNV
          BAgTCExBIFJJTOpBMREwDwYDVQQHDAhMTOdST8KlTzEgMB4GA1UEChMXVU5JVkVSUOlEQUQgREUg
          TEEgUklPSkExDDAKBgNVBAsTA1BGQzEfMBOGA1UEAwwWScKlSUdPIExFw6BOIFNBTUFOSUVHTzAe
          Fw0xMTA2MTUwOTEONDlaFw0xMTA5MTMwOTEONDlaMIGEMQswCQYDVQQGEwJFUzERMA8GA1UECBMI
          TEEgUklPSkExETAPBgNVBAcMCExPR1JPwqVPMSAwHgYDVQQKExdVTklWRVJTSURBRCBERSBMQSBS
          SU9KQTEMMAoGA1UECxMDUEZDMR8wHQYDVQQDDBZJwqVJRO8gTEXDoE4gUOFNQU5JRUdPMIGfMA0G
          CSqGSIb3DQEBAQUAA4GNADCBiQKBgQChePFNVCIfphFlyXQ9BysiR5BfXIuv3AnAK80Fuw4tTFwC
          nVUjJeGnkUYQO32oUu+fEBK8WsEqjeH8A7zrHTRQjfYZWyuGWrM8gJXOa/POMROPm/c/H8b5a6Nx
          1/+zLwR0tYkqLI2xqDOFII2RwK5L2yGeV4T4y8i3h1UOOFTSEwIDAQABMA0GCSqGSIb3DQEBBQUA
          A4GBAIDOvAAdOCaTpy+83bGB2KmngMJrNxxWDpAi5LGFrN8iCShmbTpIeIbYBUAaBpZtdhOnhq4n
          wD5QOENSFipQcdH5GEpPM9Rquy6xMwfda9EU5UfOSEmbk4fK2vaIOVjynpQsJ9P99enO2smQlyvw
          /hBa7Xacz6qDut8ghUeuV5Js
        </ds:X509Certificate>
      </ds:X509Data>
      <ds:KeyValue>
        <ds:RSAKeyValue>
          <ds:Modulus>
            oXjxTVQiH6YRZcl0PQcrIkeQX1yLr9wJwCvNBbsOLUxcAp1VIyXhp5FGEDt9qFLvnxASvFrBKo3h
            /AO86x00UI32GVsrhlqzPICVzmvz9DETj5v3Px/G+Wujcdf/sy8EdLWJKiyNsagzhSCNkcCuS9sh
            nleE+MvIt4dVNDhUOhM=
          </ds:Modulus>
          <ds:Exponent>AQAB</ds:Exponent>
        </ds:RSAKeyValue>
      </ds:KeyValue>
    </ds:KeyInfo>
  </ds:Signature>
</pnml>
```

In the generated **<Signature>** a XMLSignature XPath Filter tag has appeared as a new **<Transform>**

```
<ds:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
  <dsig-xpath:XPath
      xmlns:dsig-xpath="http://www.w3.org/2002/06/xmldsig-filter2"
      Filter="union">
    /
  </dsig-xpath:XPath>
</ds:Transform>
```

It is easy to see the behaviour of this element: the union of "/" XPath expression, that is to say, the union of the entire document.

The set of nodes returned by this expression is

```
<pnml>
  <net id="myNet" type="http://www.pnml.org/version-2009/grammar/ptnet">
    <name>
      <text> My new net </text>
    </name>
    <page id="page1">
      <subnet id="sn1">
        <interface id="sn1-interface">
        <gate action="input" id="igp1" type="place"></gate>
        <gate action="input" id="igp2" type="place"></gate>
        <gate action="output" id="ogt1" type="transition">
          <inscription>
          <text> 2 </text>
          </inscription>
        </gate>
        </interface>
        <content id="sn1-content">
        <place id="p2"></place>
        <place id="p3"></place>
        <transition id="t3"></transition>
        <arc id="a2" source="igp2" target="p2"></arc>
        <arc id="a3" source="igp1" target="p3"></arc>
        <arc id="a4" source="p3" target="ogt1">
          <inscription>
          <text> 2 </text>
          </inscription>
        </arc>
        <arc id="a5" source="t3" target="p3"></arc>
        <arc id="a6" source="p2" target="t3"></arc>
        </content>
      </subnet>
      <place id="p1"></place>
      <transition id="t1"></transition>
      <transition id="t2"></transition>
      <arc id="a1" source="p1" target="t1">
        <inscription>
```

```
        <text> 3 </text>
        </inscription>
      </arc>
      <arc id="a2" source="t1" target="igp2"></arc>
      <arc id="a3" source="t1" target="igp1"></arc>
      <arc id="a4" source="ogt1" target="t2">
        <inscription>
        <text> 2 </text>
        </inscription>
      </arc>
      <arc id="a7" source="t2" target="p1"></arc>
    </page>
  </net>
</pnml>
```

It is the full document (without the first row xml definition) node set.

Other important configuration in this work is the signing of a concrete subnet. In this case, it is a little different as in XMLEncryption. Remember that in XMLEncription, if I want to mask a subnet I don't process the `<subnet>` tag but the `subnet/content` this is because the interface has to be visible. But in a signature I want to sign the complete subnet, including the interface. Suppose that this subnet has `id="sn1"`. The XPath expression that represents it is:

```
/net/page/subnet[@id="sn1"]
```

The node set in this case is

```
<subnet id="sn1">
  <interface id="sn1-interface">
    <gate action="input" id="igp1" type="place"/>
    <gate action="input" id="igp2" type="place"/>
    <gate action="output" id="ogt1" type="transition">
      <inscription>
        <text> 2 </text>
      </inscription>
    </gate>
  </interface>
  <content id="sn1-content">
    <place id="p2"/>
    <place id="p3"/>
    <transition id="t3"/>
    <arc id="a2" source="igp2" target="p2"/>
    <arc id="a3" source="igp1" target="p3"/>
    <arc id="a4" source="p3" target="ogt1">
      <inscription>
        <text> 2 </text>
      </inscription>
    </arc>
    <arc id="a5" source="t3" target="p3"/>
    <arc id="a6" source="p2" target="t3"/>
  </content>
</subnet>
```

and the signature result is

```
<?xml version="1.0" encoding="UTF-8"?>
<pnml>
  <net id="myNet" type="http://www.pnml.org/version-2009/grammar/ptnet">
    <name>
      <text> My new net </text>
    </name>
    <page id="page1">
      <subnet id="sn1">
        <interface id="sn1-interface">
          <gate action="input" id="igp1" type="place"/>
          <gate action="input" id="igp2" type="place"/>
          <gate action="output" id="ogt1" type="transition">
            <inscription>
              <text> 2 </text>
            </inscription>
          </gate>
        </interface>
        <content id="sn1-content">
          <place id="p2"/>
          <place id="p3"/>
          <transition id="t3"/>
          <arc id="a2" source="igp2" target="p2"/>
          <arc id="a3" source="igp1" target="p3"/>
          <arc id="a4" source="p3" target="ogt1">
            <inscription>
              <text> 2 </text>
            </inscription>
          </arc>
          <arc id="a5" source="t3" target="p3"/>
          <arc id="a6" source="p2" target="t3"/>
        </content>
      </subnet>
      <place id="p1"/>
      <transition id="t1"/>
      <transition id="t2"/>
      <arc id="a1" source="p1" target="t1">
        <inscription>
          <text> 3 </text>
        </inscription>
      </arc>
      <arc id="a2" source="t1" target="igp2"/>
      <arc id="a3" source="t1" target="igp1"/>
      <arc id="a4" source="ogt1" target="t2">
        <inscription>
          <text> 2 </text>
        </inscription>
      </arc>
      <arc id="a7" source="t2" target="p1"/>
    </page>
  </net>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <ds:Reference URI="">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
          <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
          <ds:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
            <dsig-xpath:XPath xmlns:dsig-xpath="http://www.w3.org/2002/06/xmldsig-filter2" Filter="intersect">
              /pnml/net/page/subnet[@id="sn1"]
            </dsig-xpath:XPath>
          </ds:Transform>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>prCzhLgTCZ1ck6MjQnFy6cASCZw=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>
      QoO7mQmGBFTg2UxgiZnzlsnKi8V477JC0v12JPItL53zIOCpjhOwLoyxENl6v8lCLoJ9WwFHlBKk
      r3GdqrgZimNXMUjwR4zkd9FVNcIrn85DuRjHA/zDwSuPMq9w0N5AO7c0xJ24uvn9+zpbQxfblYTb
      kiy08+S0pqczU/bv5+g=
    </ds:SignatureValue>
    <ds:KeyInfo>
```

```
      <ds:X509Data>
        <ds:X509Certificate>
          MIICgTCCAeqgAwIBAgIETfh4CTANBgkqhkiG9w0BAQUFADCBhDELMAkGA1UEBhMCRVMxETAPBgNV
          BAgTCExBIFJJT0pBMREwDwYDVQQHDAhMT0dST8KlTzEgMB4GA1UEChMXVU5JVkVSU0lEQUQgREUg
          TEEgUklPSkExDDAKBgNVBAsTA1BGQzEfMB0GA1UEAwwWScKlSUdPIExFw6BOIFNBTUFOSUVHTzAe
          Fw0xMTA2MTUwOTE0NDlaFw0xMTA5MTMwOTE0NDlaMIGEMQswCQYDVQQGEwJFUzERMA8GA1UECBMI
          TEEgUklPSkExETAPBgNVBAcMCExPR1JPwqVPMSAwHgYDVQQKExdVTklWRVJTSURBRCBERSBMQSBS
          SU9KQTEMMAoGA1UECxMDUEZDMR8wHQYDVQQDDBZJwqVJRO8gTEXDoE4gUOFNQU5JRUdPMIGfMA0G
          CSqGSIb3DQEBAQUAA4GNADCBiQKBgQChePFNVCIfphFlyXQ9BysiR5BfXIuv3AnAK8OFuw4tTFwC
          nVUjJeGnkUYQO32oUu+fEBK8WsEqjeH8A7zrHTRQjfYZWyuGWrM8gJXOa/POMROPm/c/H8b5a6Nx
          1/+zLwR0tYkqLI2xqDOFII2RwK5L2yGeV4T4y8i3h1UOOFTSEwIDAQABMA0GCSqGSIb3DQEBBQUA
          A4GBAIDOvAAdOCaTpy+83bGB2KmngMJrNxxWDpAi5LGFrN8iCShmbTpIeIbYBUAaBpZtdhOnhq4n
          wD5QOENSFipQcdH5GEpPM9Rquy6xMwfda9EU5UfOSEmbk4fK2vaIOVjynpQsJ9P99enO2smQlyvw
          /hBa7Xacz6qDut8ghUeuV5Js
        </ds:X509Certificate>
      </ds:X509Data>
      <ds:KeyValue>
        <ds:RSAKeyValue>
          <ds:Modulus>
            oXjxTVQiH6YRZcl0PQcrIkeQX1yLr9wJwCvNBbsOLUxcAp1VIyXhp5FGEDt9qFLvnxASvFrBKo3h
            /AO86x00UI32GVsrhlqzPICVzmvz9DETj5v3Px/G+Wujcdf/sy8EdLWJKiyNsagzhSCNkcCuS9sh
            nleE+MvIt4dVNDhUOhM=
          </ds:Modulus>
          <ds:Exponent>AQAB</ds:Exponent>
        </ds:RSAKeyValue>
      </ds:KeyValue>
    </ds:KeyInfo>
  </ds:Signature>
</pnml>
```

In this case, the Transform associated to the XPath expression is

```
<ds:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
  <dsig-xpath:XPath
      xmlns:dsig-xpath="http://www.w3.org/2002/06/xmldsig-filter2"
      Filter="intersect">
    /pnml/net/page/subnet[@id="sn1"]
  </dsig-xpath:XPath>
</ds:Transform>
```

As we can see, the resultant node set is the intersection of the entire document and the nodes returned by `/pnml/net/page/subnet[@id="sn1"]`, that is, the nodes returned by `/pnml/net/page/subnet[@id="sn1"]`.

5.3.4 Examples

5.3.4.1 Signing all the subnets of a Petri net

Other important configuration is to sign all the subnets defined in a PNML file. Let's take the example Petri net of the section 5.2.3.1. The goal is to sign all the subnets... //TODO

5.3.4.2 Signing several different subnets

In this part a configuration is to sign several subnets (but not all) defined in a PNML file is explained. Taking the example of the section 5.2.3.1

//TODO

5.4 Complete security

In this section I am going to explain several important questions. Until now, I have described the two security operation separately: hiding and signing. I want some parts to be hidden and other parts to be signed. But, what would happen if I want to hide and sign the same parts? In this case, the result is different depending on the order selected. Let's see the differences.

Suppose first cipher and then sign. If this order is applied, the cipher has no problem, but the sign has a obvious but important detail: the signed content is encrypted itself. What does it mean? Well, you don't see what you are signing. The implications are simple: you don't know what you are signing. This order is useful in order to guarantee the integrity, but you have to be sure that what you are signing is exactly what you want.

The other order is: first signing and then ciphering. In this case, signing has no problem. The special characteristic is the ciphering. As explained before in this chapter, after signing, appears a new tag `<Signature>` appended to the original content. Then, if I encrypt the signed content, this tag is still visible but the signature cannot be verified, because the encrypting process replace the original content, so there is a modification of the signed data and it is detected by the signature, avoiding a correct verification of it.

The selected order depends on the responsible of each part. Anyway, there is a W3C recommendation that describes a XMLSignature decryption transform [83] that permits to differentiate between ciphered content that were encrypted before signing (and must not be decrypted) and ciphered content that were encrypted after signing (and must be decrypted). So the XML applications are able to interpret the correct way to validate the signature for the signature to validate the signature. As it is a particular case of configuration, and my intention is explain the general guidelines, I am not going to explain it here. Again, the responsible of the net/subnet is who has to decide the concrete configuration, but maintaining the general rules that I explain in this thesis.

5.5 Conclusions

As we have seen, one of the applications of subnetting Petri nets and represent them in PNML is that I can apply standard processes based on XML. In this case, once defined and represented a subnet in PNML, I have applied XMLEncryption in order to hide the internal structure of the subnet and XMLSignature for signing the subnet.

The use of standard and widely extended technologies like XMLEncryption and XMLSignature enables that everybody has access to them, so the processes explained here are perfectly usable.

One important conclusion in the encryption section is that several subnets in the same net can be ciphered with different options. This is used, for example, for one receiver to decrypt the subnet addressed to him but no other one.

Other important use is that with a PNML file with encrypted subnets, I can replace one ciphered subnet by other ciphered subnet without the necessity of decrypt any of them: we can work with encrypted subnets if the interfaces are the same. With the explained in this chapter, the privacy of parts of a Petri net is guarantied.

In XMLSignature, integrity, authentication and non repudiation is assured. A big difference with XMLEncryption is that the signature is attached to the original file, instead of replace part of the net. In this case I have used enveloped signature, but if we use the detached way, the signature is in other file, separating signature and signed content. In signature it makes no sense replace one signature by another, but we can have as many signatures as we want, including different signatures of the same contents.

Using both technologies XMLEncryption and XMLSignature together we can reach very high security levels, but we have to be careful with the order of the processes, because depending on the selected order, the acquired properties will be different.

The last conclusion, but very important one is that the responsible of the net has to choose the concrete configurations of XMLEncryption and XMLSignature, the transforms and the order of the actions in order to achieve his particular objectives.

Chapter 6

Conclusions

Intentar que sea un resumen de todo lo tratado, haciendo hincapie en lo aportado y en futuras lineas de investigacion. Que no sea corta. Lo suficientemente larga para que alguien que se lea la tesis pueda recordarla completamente

Throughout this paper we have presented Petri nets with definitions and basic properties. From this initial presentation, have been building a series of elements as a basis for further investigation. We defined subnets, subnets classifications have been studied, we have defined front-ends (interfaces) for those subnets, etc.. From this point is possible a further study of these subnets (their properties, utilities,....)

The contribution of this research is to establish the basis for the methodological study of hiding parts of Petri nets. We present the definitions and some basic properties and from here is to complete it by preparing a doctoral thesis.

Chapter 7

Ejemplos Latex

7.1 Codigo XML

Para poner codigo xml dentro del documento

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="points">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="point">
          <xs:complexType>
            <xs:attribute name="x" type="xs:unsignedShort" use="required"/>
            <xs:attribute name="y" type="xs:unsignedShort" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Un poco mas pequeno

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="points">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="point">
          <xs:complexType>
            <xs:attribute name="x" type="xs:unsignedShort" use="required" />
            <xs:attribute name="y" type="xs:unsignedShort" use="required" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

7.2 varias columnas

Para poner texto en varias columnas

Datos en dos columnas fndio fpeowu fpieuwpu piud pqeudp oquewp duqwipud pqwudpqwiodu pqwidu pqwidu pqwiud piqwud pqwu dpqwoud pqwod qpwodu qwpdu qwpd upou pqowdu wpqo po uqwpod u

7.3 Figuras

Esta figura se pone exactamente donde se define. No se deja a latex elegir el sitio

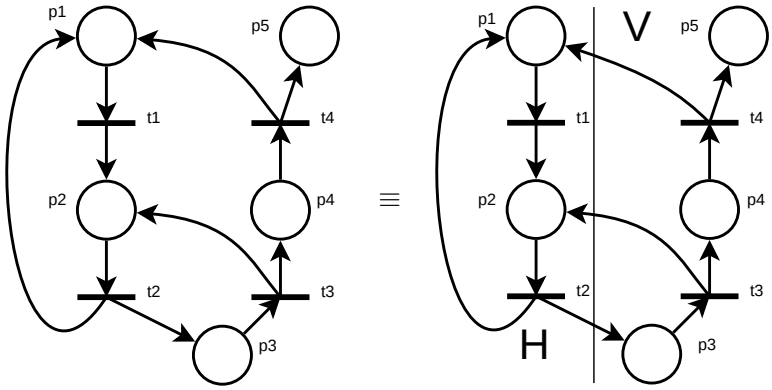


FIGURE 7.1: figura con dos partes y que se pone exactamente donde se define

Sin embargo esta otra la elige latex

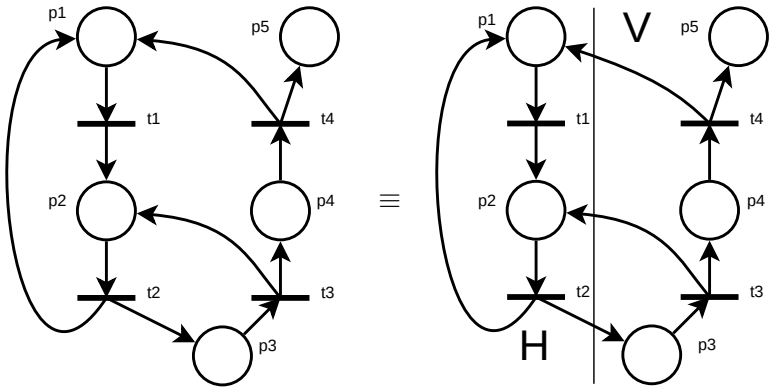


FIGURE 7.2: figura con dos partes general

Appendix A

PNML grammar

The official PNML grammar is available in www.pnml.org. It is written in RELAX NG format. This are the main parts that I use in this work.

A.1 RELAX NG implementation of PNML Core Model

```
<?xml version="1.0" encoding="UTF-8"?>

<grammar ns="http://www.pnml.org/version-2009/grammar/pnml"
  xmlns="http://relaxng.org/ns/structure/1.0"
  xmlns:a="http://relaxng.org/ns/compatibility/annotations/1.0"
  datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">

  <a:documentation>
    Petri Net Markup Language (PNML) schema.
    RELAX NG implementation of PNML Core Model.

    File name: pnmlcoremodel.rng
    Version: 2009
    (c) 2001-2009
    Michael Weber,
    Ekkart Kindler,
    Christian Stehno,
    Lom Hillah (AFNOR)
    Revision:
    July 2008 - L.H
  </a:documentation>

  <include href="http://www.pnml.org/version-2009/grammar/anyElement.rng"/>

  <start>
    <ref name="pnml.element"/>
  </start>

  <define name="pnml.element">
```

```
    <element name="pnml">
      <a:documentation>
        A PNML document consists of one or more Petri nets.
        It has a version.
      </a:documentation>
      <oneOrMore>
        <ref name="pnml.content"/>
      </oneOrMore>
    </element>
  </define>

  <define name="pnml.content">
    <ref name="net.element"/>
  </define>

  <define name="net.element">
    <element name="net">
      <a:documentation>
        A net has a unique identifier (id) and refers to
        its Petri Net Type Definition (PNTD) (type).
      </a:documentation>
      <ref name="identifier.content"/>
      <ref name="nettype.uri"/>
      <a:documentation>
        The sub-elements of a net may occur in any order.
        A net consists of at least a top-level page which
        may contain several objects. A net may have a name,
        other labels (net.labels) and tool specific information in any order.
      </a:documentation>
      <interleave>
        <optional>
          <ref name="Name"/>
        </optional>
        <ref name="net.labels"/>
        <oneOrMore>
          <ref name="page.content"/>
        </oneOrMore>
        <zeroOrMore>
          <ref name="toolspecific.element"/>
        </zeroOrMore>
      </interleave>
    </element>
  </define>

  <define name="identifier.content">
    <a:documentation>
      Identifier (id) declaration shared by all objects in any PNML model.
    </a:documentation>
    <attribute name="id">
      <data type="ID"/>
    </attribute>
  </define>

  <define name="nettype.uri">
    <a:documentation>
```

```
      The net type (nettype.uri) of a net should be redefined in the grammar
      for a new Petri net Type.
      An example of such a definition is in ptnet.pntd, the grammar
      for P/T Nets. The following value is a default.
    </a:documentation>
    <attribute name="type">
      <value>http://www.pnml.org/version-2009/grammar/pnmlcoremodel</value>
    </attribute>
  </define>

  <define name="net.labels">
    <a:documentation>
      A net may have unspecified many labels. This pattern should be used
      within a PNTD to define the net labels.
    </a:documentation>
    <empty/>
  </define>

  <define name="basicobject.content">
    <a:documentation>
      Basic contents for any object of a PNML model.
    </a:documentation>
    <interleave>
      <optional>
        <ref name="Name"/>
      </optional>
      <zeroOrMore>
        <ref name="toolspecific.element"/>
      </zeroOrMore>
    </interleave>
  </define>


  <define name="page.content">
    <a:documentation>
      A page has an id. It may have a name and tool specific information.
      It may also have graphical information. It can also have many arbitrary labels.
      Note: according to this definition, a page may contain other pages.
      All these sub-elements may occur in any order.
    </a:documentation>
    <element name="page">
      <ref name="identifier.content"/>
      <interleave>
        <ref name="basicobject.content"/>
        <ref name="page.labels"/>
        <zeroOrMore>
          <ref name="netobject.content"/>
        </zeroOrMore>
        <optional>
          <element name="graphics">
            <ref name="pagegraphics.content"/>
          </element>
        </optional>
      </interleave>
    </element>
```

```
</define>

<define name="netobject.content">
  <a:documentation>
    A net object is either a page, a node or an arc.
    A node is a place or a transition, a reference place of
    a reference transition.
  </a:documentation>
  <choice>
    <ref name="page.content"/>
    <ref name="place.content"/>
    <ref name="transition.content"/>
    <ref name="refplace.content"/>
    <ref name="reftrans.content"/>
    <ref name="arc.content"/>
  </choice>
</define>

<define name="page.labels">
  <a:documentation>
    A page may have unspecified many labels. This pattern should be used
    within a PNTD to define new labels for the page concept.
  </a:documentation>
  <empty/>
</define>

<define name="place.content">
  <a:documentation>
    A place may have several labels (place.labels) and the same content
    as a node.
  </a:documentation>
  <element name="place">
    <ref name="identifier.content"/>
    <interleave>
      <ref name="basicobject.content"/>
      <ref name="place.labels"/>
      <ref name="node.content"/>
    </interleave>
  </element>
</define>

<define name="place.labels">
  <a:documentation>
    A place may have arbitrary many labels. This pattern should be used
    within a PNTD to define the place labels.
  </a:documentation>
  <empty/>
</define>

<define name="transition.content">
  <a:documentation>
    A transition may have several labels (transition.labels) and the same
    content as a node.
  </a:documentation>
  <element name="transition">
```

```
      <ref name="identifier.content"/>
      <interleave>
        <ref name="basicobject.content"/>
        <ref name="transition.labels"/>
        <ref name="node.content"/>
      </interleave>
    </element>
  </define>

  <define name="transition.labels">
    <a:documentation>
      A transition may have arbitrary many labels. This pattern should be
      used within a PNTD to define the transition labels.
    </a:documentation>
    <empty/>
  </define>

  <define name="node.content">
    <a:documentation>
      A node may have graphical information.
    </a:documentation>
    <optional>
      <element name="graphics">
        <ref name="nodegraphics.content"/>
      </element>
    </optional>
  </define>

  <define name="reference">
    <a:documentation>
      Here, we define the attribute ref including its data type.
      Modular PNML will extend this definition in order to change
      the behavior of references to export nodes of module instances.
    </a:documentation>
    <attribute name="ref">
      <data type="IDREF"/>
    </attribute>
  </define>

  <define name="refplace.content">
    <a:documentation>
      A reference place is a reference node.
    </a:documentation>
    <a:documentation>
      Validating instruction:
      - _ref_ MUST refer to _id_ of a reference place or of a place.
      - _ref_ MUST NOT refer to _id_ of its reference place element.
      - _ref_ MUST NOT refer to a cycle of reference places.
    </a:documentation>
    <element name="referencePlace">
      <ref name="refnode.content"/>
    </element>
  </define>

  <define name="reftrans.content">
```

```
    <a:documentation>
      A reference transition is a reference node.
    </a:documentation>
    <a:documentation>
      Validating instruction:
      - The reference (ref) MUST refer to a reference transition or to a
      transition.
      - The reference (ref) MUST NOT refer to the identifier (id) of its
      reference transition element.
      - The reference (ref) MUST NOT refer to a cycle of reference transitions.
    </a:documentation>
    <element name="referenceTransition">
      <ref name="refnode.content"/>
    </element>
  </define>

  <define name="refnode.content">
    <a:documentation>
      A reference node has the same content as a node.
      It adds a reference (ref) to a (reference) node.
    </a:documentation>
    <ref name="identifier.content"/>
    <ref name="reference"/>
    <ref name="basicobject.content"/>
    <ref name="node.content"/>
  </define>

  <define name="arc.content">
    <a:documentation>
      An arc has a unique identifier (id) and
      refers both to the node's id of its source and
      the node's id of its target.
      In general, if the source attribute refers to a place,
      then the target attribute refers to a transition and vice versa.
    </a:documentation>
    <element name="arc">
      <ref name="identifier.content"/>
      <attribute name="source">
        <data type="IDREF"/>
      </attribute>
      <attribute name="target">
        <data type="IDREF"/>
      </attribute>
      <a:documentation>
        The sub-elements of an arc may occur in any order.
        An arc may have a name, graphical and tool specific information.
        It may also have several labels.
      </a:documentation>
      <interleave>
        <optional>
          <ref name="Name"/>
        </optional>
        <ref name="arc.labels"/>
        <optional>
          <element name="graphics">
```

```
          <ref name="edgegraphics.content"/>
        </element>
      </optional>
      <zeroOrMore>
        <ref name="toolspecific.element"/>
      </zeroOrMore>
    </interleave>
  </element>
</define>

<define name="arc.labels">
  <a:documentation>
    An arc may have arbitrary many labels. This pattern should be used
    within a PNTD to define the arc labels.
  </a:documentation>
  <empty/>
</define>

<define name="pagegraphics.content">
  <a:documentation>
    A page graphics is actually a node graphics
  </a:documentation>
  <ref name="nodegraphics.content"/>
</define>

<define name="nodegraphics.content">
  <a:documentation>
    The sub-elements of a node's graphical part occur in any order.
    At least, there may be one position element.
    Furthermore, there may be a dimension, a fill, and a line element.
  </a:documentation>
  <interleave>
    <ref name="position.element"/>
    <optional>
      <ref name="dimension.element"/>
    </optional>
    <optional>
      <ref name="fill.element"/>
    </optional>
    <optional>
      <ref name="line.element"/>
    </optional>
  </interleave>
</define>

<define name="edgegraphics.content">
  <a:documentation>
    The sub-elements of an arc's graphical part occur in any order.
    There may be zero or more position elements.
    Furthermore, there may be a line element.
  </a:documentation>
  <interleave>
    <zeroOrMore>
      <ref name="position.element"/>
    </zeroOrMore>
```

```
    <optional>
      <ref name="line.element"/>
    </optional>
  </interleave>
</define>

<define name="simpletext.content">
  <a:documentation>
    This definition describes the contents of simple text labels
    without graphics.
  </a:documentation>
  <optional>
    <element name="text">
      <a:documentation>
        A text should have a value.
        If not, then there must be a default.
      </a:documentation>
      <text/>
    </element>
  </optional>
</define>

<define name="annotationstandard.content">
  <a:documentation>
    The definition annotationstandard.content describes the
    standard contents of an annotation.
    Each annotation may have graphical or tool specific information.
  </a:documentation>
  <interleave>
    <optional>
      <element name="graphics">
        <ref name="annotationgraphics.content"/>
      </element>
    </optional>
    <zeroOrMore>
      <ref name="toolspecific.element"/>
    </zeroOrMore>
  </interleave>
</define>

<define name="simpletextlabel.content">
  <a:documentation>
    A simple text label is an annotation to a net object containing
    arbitrary text.
    Its sub-elements occur in any order.
    A simple text label behaves like an attribute to a net object.
    Furthermore, it contains the standard annotation contents which
    basically defines the graphics of the text.
  </a:documentation>
  <interleave>
    <ref name="simpletext.content"/>
    <ref name="annotationstandard.content"/>
  </interleave>
</define>
```

```
<define name="Name">
  <a:documentation>
    Label definition for a user given name of an
    element.
  </a:documentation>
  <element name="name">
    <ref name="simpletextlabel.content"/>
  </element>
</define>

<define name="annotationgraphics.content">
  <a:documentation>
    An annotation's graphics part requires an offset element describing
    the offset the center point of the surrounding text box has to
    the reference point of the net object on which the annotation occurs.
    Furthermore, an annotation's graphic element may have a fill, a line,
    and font element.
  </a:documentation>
  <ref name="offset.element"/>
  <interleave>
    <optional>
      <ref name="fill.element"/>
    </optional>
    <optional>
      <ref name="line.element"/>
    </optional>
    <optional>
      <ref name="font.element"/>
    </optional>
  </interleave>
</define>

<define name="position.element">
  <a:documentation>
    A position element describes Cartesian coordinates.
  </a:documentation>
  <element name="position">
    <ref name="coordinate.attributes"/>
  </element>
</define>

<define name="offset.element">
  <a:documentation>
    An offset element describes Cartesian coordinates.
  </a:documentation>
  <element name="offset">
    <ref name="coordinate.attributes"/>
  </element>
</define>

<define name="coordinate.attributes">
  <a:documentation>
    The coordinates are decimal numbers and refer to an appropriate
    xy-system where the x-axis runs from left to right and the y-axis
    from top to bottom.
```

```
    </a:documentation>
    <attribute name="x">
      <data type="decimal"/>
    </attribute>
    <attribute name="y">
      <data type="decimal"/>
    </attribute>
  </define>

  <define name="dimension.element">
    <a:documentation>
      A dimension element describes the width (x coordinate) and height
      (y coordinate) of a node.
      The coordinates are actually positive decimals.
    </a:documentation>
    <element name="dimension">
      <attribute name="x">
        <ref name="positiveDecimal.content"/>
      </attribute>
      <attribute name="y">
        <ref name="positiveDecimal.content"/>
      </attribute>
    </element>
  </define>

  <define name="positiveDecimal.content" ns="http://www.w3.org/2001/XMLSchema-datatypes">
    <a:documentation>
      Definition of a restricted positive decimals domain with a total digits
      number of 4 and 1 fraction digit. Ranges from 0 to 999.9
    </a:documentation>
    <data type='decimal'>
      <param name='totalDigits'>4</param>
      <param name='fractionDigits'>1</param>
      <param name='minExclusive'>0</param>
    </data>
  </define>

  <define name="fill.element">
    <a:documentation>
      A fill element describes the interior colour, the gradient colour,
      and the gradient rotation between the colors of an object.  If an
      image is available the other attributes are ignored.
    </a:documentation>
    <element name="fill">
      <optional>
        <attribute name="color">
          <ref name="color.type"/>
        </attribute>
      </optional>
      <optional>
        <attribute name="gradient-color">
          <ref name="color.type"/>
        </attribute>
      </optional>
      <optional>
```

```
      <attribute name="gradient-rotation">
        <choice>
          <value>vertical</value>
          <value>horizontal</value>
          <value>diagonal</value>
        </choice>
      </attribute>
    </optional>
    <optional>
      <attribute name="image">
        <data type="anyURI"/>
      </attribute>
    </optional>
  </element>
</define>

<define name="line.element">
  <a:documentation>
    A line element describes the shape, the colour, the width, and the
    style of an object.
  </a:documentation>
  <element name="line">
    <optional>
      <attribute name="shape">
        <choice>
          <value>line</value>
          <value>curve</value>
        </choice>
      </attribute>
    </optional>
    <optional>
      <attribute name="color">
        <ref name="color.type"/>
      </attribute>
    </optional>
    <optional>
      <attribute name="width">
        <ref name="positiveDecimal.content"/>
      </attribute>
    </optional>
    <optional>
      <attribute name="style">
        <choice>
          <value>solid</value>
          <value>dash</value>
          <value>dot</value>
        </choice>
      </attribute>
    </optional>
  </element>
</define>

<define name="color.type">
  <a:documentation>
    This describes the type of a color attribute. Actually, this comes
```

```
      from the CSS2 (and latest versions) data type system.
    </a:documentation>
    <text/>
  </define>

  <define name="font.element">
    <a:documentation>
      A font element describes several font attributes, the decoration,
      the alignment, and the rotation angle of an annotation's text.
      The font attributes (family, style, weight, size) should be conform
      to the CSS2 and latest versions data type system.
    </a:documentation>
    <element name="font">
      <optional>
        <attribute name="family">
          <text/>  <!-- actually, CSS2 and latest versions font-family -->
        </attribute>
      </optional>
      <optional>
        <attribute name="style">
          <text/>  <!-- actually, CSS2 and latest versions font-style -->
        </attribute>
      </optional>
      <optional>
        <attribute name="weight">
          <text/>  <!-- actually, CSS2 and latest versions font-weight -->
        </attribute>
      </optional>
      <optional>
        <attribute name="size">
          <text/>  <!-- actually, CSS2 and latest versions font-size -->
        </attribute>
      </optional>
      <optional>
        <attribute name="decoration">
          <choice>
            <value>underline</value>
            <value>overline</value>
            <value>line-through</value>
          </choice>
        </attribute>
      </optional>
      <optional>
        <attribute name="align">
          <choice>
            <value>left</value>
            <value>center</value>
            <value>right</value>
          </choice>
        </attribute>
      </optional>
      <optional>
        <attribute name="rotation">
          <data type="decimal"/>
        </attribute>
```

```
      </optional>
    </element>
  </define>

  <define name="toolspecific.element">
    <a:documentation>
      The tool specific information refers to a tool and its version.
      The further substructure is up to the tool.
    </a:documentation>
    <element name="toolspecific">
      <attribute name="tool">
        <text/>
      </attribute>
      <attribute name="version">
        <text/>
      </attribute>
      <zeroOrMore>
        <ref name="anyElement"/>
      </zeroOrMore>
    </element>
  </define>

</grammar>
```

Listing A.1: RELAX NG implementation of PNML Core Model

A.2 RELAX NG implementation of Petri Net Type Definition for Place/Transition nets

```
<?xml version="1.0" encoding="UTF-8"?>

<grammar ns="http://www.pnml.org/version-2009/grammar/pnml"
  xmlns="http://relaxng.org/ns/structure/1.0"
  xmlns:a="http://relaxng.org/ns/compatibility/annotations/1.0">

  <a:documentation>
    RELAX NG implementation of Petri Net Type Definition for Place/Transition nets.
    This PNTD re-defines the value of nettype.uri for P/T nets.

    File name: ptnet.pntd
    Version: 2009
    (c) 2007-2009
    Lom Hillah (AFNOR)
    Revision:
    July 2008 - L.H
  </a:documentation>

  <a:documentation>
    The PT Net type definition.
    This document also declares its namespace.
    All labels of this Petri net type come from the Conventions document.
    The use of token graphics as tool specific feature is possible.
  </a:documentation>

  <include href="http://www.pnml.org/version-2009/grammar/conventions.rng"/>

  <!--
  <include href="http://www.pnml.org/version-2009/grammar/pnmlextensions.rng"/>
    We do not need to include this, because the pnmlcoremodel.rng covers any
    toolspecific extension.
  -->

  <include href="http://www.pnml.org/version-2009/grammar/pnmlcoremodel.rng"/>

  <define name="nettype.uri" combine="choice">
    <a:documentation>
      The URI value for the net type attribute,
      declaring the type of P/T nets.
    </a:documentation>
    <attribute name="type">
      <value>http://www.pnml.org/version-2009/grammar/ptnet</value>
    </attribute>
  </define>


  <define name="PTMarking">
    <a:documentation>
      Label definition for initial marking in nets like P/T-nets.
      <contributed>Michael Weber</contributed>
      <date>2003-06-16</date>
```

```
      <reference>
        W. Reisig: Place/transition systems. In: LNCS 254. 1987.
      </reference>
    </a:documentation>
    <element name="initialMarking">
      <ref name="nonnegativeintegerlabel.content"/>
    </element>
  </define>

  <define name="PTArcAnnotation">
    <a:documentation>
      Label definition for arc inscriptions in P/T-nets.
      <contributed>Michael Weber, AFNOR</contributed>
      <date>2003-06-16</date>
      <reference>
        W. Reisig: Place/transition systems. In: LNCS 254. 1987.
      </reference>
    </a:documentation>
    <element name="inscription">
      <ref name="positiveintegerlabel.content"/>
    </element>
  </define>

  <define name="place.labels" combine="interleave">
    <a:documentation>
      A place of a P/T net may have an initial marking.
    </a:documentation>
    <optional><ref name="PTMarking"/></optional>
  </define>

  <define name="arc.labels" combine="interleave">
    <a:documentation>
      An arc of a P/T net may have an inscription.
    </a:documentation>
    <optional><ref name="PTArcAnnotation"/></optional>
  </define>

</grammar>
```

Listing A.2: RELAX NG implementation of PNTD for Place/Transition nets

Bibliography

[1] E. Jiménez Macías and M. Pérez de la Parte. Simulation and optimization of logistic and production systems using discrete and continuous petri nets. Simulation, 80 (3):143–152, 2004.

[2] T. Guasch, M A. Piera, J. Casanovas, and J. Figueras. Modelado y Simulación. Aplicación a procesos logísticos de fabricación y servicios. Edicions UPC, Barcelona, Spain, 2002.

[3] K Jensen and L.M Kristensen. Coloured Petri Nets: Modelling and validation of concurrent systems. 2009. doi: 10.1007/b95112. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84892340728&partnerID=40&md5=362ee3ffac5963e6564913484021b209. cited By 176.

[4] I. León Samaniego. Seguridad y protección en envío y almacenamiento de datos. firmado y cifrado. aplicación a redes de petri y gestión de residuos con e3l. Master's thesis, Universidad de La Rioja. Logroño, Spain, 2011.

[5] SA DRUZHININ, VA; YUDITSKII. Construction of well-formed petri nets from standard subnets. AUTOMATION AND REMOTE CONTROL, 53(12):1922–1927, 1992.

[6] H.M.A. Fahmy. Analysis of petri nets by partitioning: Splitting transitions. Theoretical Computer Science, 77(3):321–330, 1990. doi: 10.1016/0304-3975(90)90174-G. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0025594586&partnerID=40&md5=39dd67dc9ab7d1b847a7b9f17b7592bf. cited By 0.

[7] Hossam Mahmoud Ahmad Fahmy. Analysis of petri nets by partitioning: splitting places or transitions. International Journal of Computer Mathematics, 48(3-4):127–148, 1993. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0027813981&partnerID=40&md5=8b77f48d840bf23442b2ed12dbf34d73. cited By 1.

[8] I Suzuki and T Murata. A method for stepwise refinement and abstraction of petri nets. Journal of Computer and System Sciences, 27(1):51–76, 1983. doi: 10.1016/0022-0000(83)90029-6. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0020795351&partnerID=40&md5=7afaa0ab0c08ed65aa92cfca555f4fd6. cited By 125.

[9] R. Valette. Analysis of petri nets by stepwise refinements. Journal of Computer and System Sciences, 18(1):35–46, 1979. doi: 10.1016/0022-0000(79)90050-3. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0002367242&partnerID=40&md5=164c506f3e7f33ac9138439a5f08b63c. cited By 99.

[10] C Xia. Analysis and application of petri subnet reduction. Procs. of the IEEE, 6 (8):1662–1669, 2011.

[11] Tadao Murata. Petri nets: properties, analysis and applications. Proceedings of the IEEE, 77(4):541–580, 1989. doi: 10.1109/5.24143. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0024645936&partnerID=40&md5=43ee2546042ef10b55eed792c6aeb409. cited By 4705.

[12] M Silva. Las Redes de Petri: en la Automática y en la Informática. Ed. AC, Madrid, Spain, 1985.

[13] R. David and H. Alla. Discrete, Continuous and Hybrid Petri Nets. Springer, Berlin, Germany, 1st ed., 2004 edition, 2010.

[14] JL Peterson. Petri Net Theory and the Modeling of Systems. Prentice Hall, Englewood Clifs, NJ, 1981.

[15] M Silva. In Practice of Petri Nets in Manufacturing. Chapman and Hall, London, UK, 1993.

[16] Carl Adam Petri. Kommunication mit Automaten. PhD thesis, Technischen Hoschule Darmstadt, 1962.

[17] Carl Adam Petri. Communication with automata. Technical Report 65-377, Rome Air Devolopement Center, 1966.

[18] Carl Adam Petri. Interpretations of net theory. Technical Report 75-07, Gessellschaft fur Mathematik und Datenverarbeitung, Bonn, 1976.

[19] Carl Adam Petri and E; Smith. The pragmatic dimension of net theory. In In procs. of the 8th European workshop on Applications and Theory of Petri Nets, 2007.

[20] J.I. Latorre, E. Jiménez, M. Pérez, J. Blanco, and E. Martínez. The alternatives aggregation petri nets as a formalism to design discrete event systems. International Journal of Simulation and Process Modelling, 6(2):152–164, 2010. doi: 10.1504/IJSPM.2010.036019. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-77958113541&partnerID=40&md5=de863be257a873f1cc403539c0739439. cited By 8.

[21] L.E. Holloway, B.H. Krogh, and A. Giua. A survey of petri net methods for controlled discrete event systems. Discrete Event Dynamic Systems: Theory and Applications, 7(2):151–190, 1997. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0031118870&partnerID=40&md5=6f083862d531c836dbf57418b3cdd7ce. cited By 220.

[22] J.I. Latorre, E. Jiménez, and M. Pérez. Coloured petri nets as a formalism to represent alternative models for a discrete event system. pages 247–252, 2010. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84871324694&partnerID=40&md5=ef4fa8b5c804dd69148278197cc3587a. cited By 4.

[23] M. Silva, J. Júlvez, C. Mahulea, and C.R. Vázquez. On fluidization of discrete event models: Observation and control of continuous petri nets. Discrete Event Dynamic Systems: Theory and Applications, 21(4):427–497, 2011. doi: 10.1007/s10626-011-0116-9. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-80855132233&partnerID=40&md5=1373ddba02ca88d4580dda36b855f8ab. cited By 34.

[24] E. Jiménez, J. Júlyez, L. Recalde, and M. Silva. Relaxed continuous views of discrete event systems: Considerations on forrester diagrams and petri nets. volume 5, pages 4897–4904, 2004. doi: 10.1109/ICSMC.2004.1401307. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-15744381453&partnerID=40&md5=6e4dab53748636c40b8bffa25ef41c43. cited By 5.

[25] L. Recalde, E. Teruel, and M. Silva. Modeling and analysis of sequential processes that cooperate through buffers. IEEE Transactions on Robotics and Automation, 14(2):267–277, 1998. doi: 10.1109/70.681245. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0032048263&partnerID=40&md5=e26881591411e7172062e4c52606b95b. cited By 15.

[26] K. Jensen, L.M. Kristensen, and L. 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. doi: 10.1007/s10009-007-0038-x. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-34249933531&partnerID=40&md5=30c94bf82a91bc34487db18fa572dfca. cited By 428.

[27] L.M. Kristensen and K. Jensen. Teaching modelling and validation of concurrent systems using coloured petri nets. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5100 LNCS:19–34, 2008. doi: 10.1007/978-3-540-89287-8-2. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-58449092374&partnerID=40&md5=8cc7aa4643b2ebf3e720c7785186fe1a. cited By 0.

[28] A. Desrochers and R.Y. Al-Jaar. Applications of Petri Nets in Manufacturing Systems. Modeling, Control ad Performance Analysis. IEEE Press, New York, USA, 2010.

[29] M. Silva and E. Teruel. Petri nets for the design and operation of manufacturing systems. European Journal of Control, 3(3):182–199, 1997. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0031380765&partnerID=40&md5=301506cd728cf333b8411856d17b6e75. cited By 46.

[30] R SILVA, M; VALETTE. Petri nets and flexible manufacturing. Advances in Petri Nets, 424:374–417, 1989.

[31] M. Silva. 50 years after the phd thesis of carl adam petri: A perspective. pages 13–20, 2012. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84881038063&partnerID=40&md5=a9ebe7fa6a955a01c357a3420ed3e117. cited By 0.

[32] G. Balbo and M. Silva. Performance Models for Discrete Event Systems with Synchronizations: Formalisms and Analysis Techniques. Editorial Kronos, Zaragoza, Spain, 1998.

[33] E. Jiménez, M. Pérez, and I. Latorre. Industrial applications of petri nets: System modelling and simulation. pages 159–164, 2006. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84870231099&partnerID=40&md5=c5bd5375e5b74e79e291daac8035ee73. cited By 5.

[34] J.I. Latorre, E. Jiménez, and M. Pérez. The optimization problem based on alternatives aggregation petri nets as models for industrial discrete event systems. Simulation, 89(3):346–361, 2013. doi: 10.1177/0037549712464410. URL http:

//www.scopus.com/inward/record.url?eid=2-s2.0-84876047833&partnerID=40&md5=49380e89dca5af641e4e2108f351eeb2. cited By 12.

[35] Tadao Murata. Petri nets, marked graphs, and circuit-system theory. Circuits Syst, 11(3):2–12, 1977. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0017500512&partnerID=40&md5=aec1a2ecfd640b95fe535c63fb4a0509. cited By 10.

[36] Tadao Murata. State equation, controllability, and maximal matchings of petri nets. IEEE Transactions on Automatic Control, AC-22(3):412–416, 1977. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0017504301&partnerID=40&md5=55405f2bd4e03e2139f167355a60fef2. cited By 48.

[37] Manuel Silva. Introducing Petri nets. Chapman and Hall, 1993.

[38] Y. Lien. Termination properties of generalized petri nets. SIAM J. Comput. 5, 5 (2):251–265, 1976.

[39] Kurt Jensen. Introduction to high-level petri nets. pages 723–726, 1985. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0022285861&partnerID=40&md5=3f7da33332c8d293e110262536d36563. cited By 5.

[40] M. Silva and L. Recalde. Petri nets and integrality relaxations: A view of continuous petri net models. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 32(4):314–327, 2002. doi: 10.1109/TSMCC.2002.806063. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0036881072&partnerID=40&md5=09e69d22f01a84832ec535839d341f3a. cited By 102.

[41] V. Khomenko and M. Koutny. Branching processes of high-level petri nets. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2619:458–472, 2003. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-21144455632&partnerID=40&md5=76e0dd975ac15abbfa993536aa5fdd30. cited By 7.

[42] A.V. Ratzer, L. Wells, H.M. Lassen, M. Laursen, J.F. Qvortrup, M.S. Stissing, M. Westergaard, S. Christensen, and K. Jensen. Cpn tools for editing, simulating, and analysing coloured petri nets. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2679:450–462, 2003. URL http:

//www.scopus.com/inward/record.url?eid=2-s2.0-35248843991&partnerID=40&md5=56011fa8207463ef6e6b086c32adff61. cited By 65.

[43] L.M. Kristensen, J.B. Jorgensen, and K. Jensen. Application of coloured petri nets in system development. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 3098: 626–685, 2004. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-34250215844&partnerID=40&md5=33a9487d764a1fdb36b98fa8b22b281e. cited By 28.

[44] M. Silva and L. Recalde. On fluidification of petri nets: From discrete to hybrid and continuous models. Annual Reviews in Control, 28 (2):253–266, 2004. doi: 10.1016/j.arcontrol.2004.05.002. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-8344220389&partnerID=40&md5=986b7ccd24f7ff8dbe69fdb43b63d54f. cited By 114.

[45] J. Campos and M. Silva. Structural techniques and performance bounds of stochastic petri net models. Advances in Petri Nets, 609:352–391, 1992.

[46] L Recalde, S. Haddad, and M. Silva. Continuous petri nets: Expressive power and decidability issues. International Journal of Foundations of Computer Science, 21(2):235–256, 2010. doi: 10.1142/S0129054110007222. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-77951720712&partnerID=40&md5=5aae7a99cb5b273bc5a7b8d2fefcf8a4. cited By 11.

[47] E. Fraca, J. Júlvez, and M. Silva. Marking homothetic monotonicity and fluidization of untimed petri nets. pages 21–27, 2012. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84881057702&partnerID=40&md5=93176ea03073906e5cac7f18d7d8efc1. cited By 0.

[48] C.R. Vázquez and M. Silva. Stochastic continuous petri nets: An approximation of markovian net models. IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans, 42(3):641–653, 2012. doi: 10.1109/TSMCA.2011.2172416. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84860182487&partnerID=40&md5=47df0a4d2334d9271aa91d938afcb2be. cited By 4.

[49] C.R. Vázquez, C. Mahulea, J. Júlvez, and M. Silva. Introduction to fluid petri nets. Lecture Notes in Control and Information Sciences, 433:365–386, 2013. doi: 10.1007/978-1-4471-4276-8-18. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84866849850&partnerID=40&md5=04d88b4ee43669a4902a50003cb03e77. cited By 0.

[50] Tadao Murata. State equation, controllability, and maximal matchings of petri nets. IEEE Transactions on Automatic Control, AC-22(3):412–416, 1977. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0017504301&partnerID=40&md5=55405f2bd4e03e2139f167355a60fef2. cited By 48.

[51] J. Engelfriet. Branching processes of petri nets. Acta Informatica, 28(6): 575–591, 1991. doi: 10.1007/BF01463946. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0026191448&partnerID=40&md5=5f8e8fb4e31bd5a119c4033e91dd8582. cited By 136.

[52] M. Silva and T. Murata. B-fairness and structural b-fairness in petri net models of concurrent systems. Journal of Computer and System Sciences, 44(3):447–477, 1992. doi: 10.1016/0022-0000(92)90013-9. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0026883365&partnerID=40&md5=cf15d7c7d0d29833d9c384fb1042174b. cited By 8.

[53] L. Recalde, E. Teruel, and M. Silva. On linear algebraic techniques for liveness analysis of p/t systems. Journal of Circuits, Systems and Computers, 8(1): 223–265, 1998. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0000805834&partnerID=40&md5=7baec806fc98e6e4aef759ee0b101490. cited By 20.

[54] Q.-T. Zeng and Z.-H. Wu. Process net system of petri net. Jisuanji Xuebao/Chinese Journal of Computers, 25(12):1308–1315, 2002. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0038586362&partnerID=40&md5=4e9f182d82635987dbdb364bd02bedb3. cited By 8.

[55] E. Teruel and M. Silva. Structure theory of equal conflict systems. Theoretical Computer Science, 153(1-2):271–300, 1996. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0030574553&partnerID=40&md5=955b45543e764b5a4559b1e20352a212. cited By 31.

[56] F.-S. Hsieh. Robustness analysis of non-ordinary petri nets for flexible assembly/disassembly processes based on structural decomposition. International Journal of Control, 84(3):496–510, 2011. doi: 10.1080/00207179.2011.561443. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-79957476334&partnerID=40&md5=71a77404c1a4426281fbb5fef64689a1. cited By 11.

[57] G.S. Hura. State space representation of petri nets. Microelectronics Reliability, 24(5):865–868, 1984. doi: 10.1016/0026-2714(84)90009-X. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0021576956&partnerID=40&md5=389661c06c700381752218498a00f10a. cited By 1.

[58] N.A. Anisimov and V.L. Perchuk. Representation of exchange protocols and petri using finite sequential machine nets. Soviet journal of computer and systems sciences, 24(3):90–95, 1986. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0022712213&partnerID=40&md5=10976e98e32dd7780417db4b9ffa68f6. cited By 0.

[59] Sajal Das, V.K. Agrawal, Dilip Sarkar, L.M. Patnaik, and P.S. Goel. Reflexive incidence matrix (rim) representation of petri nets. IEEE Transactions on Software Engineering, SE-13(6):643–653, 1987. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0023366121&partnerID=40&md5=544b2cea6f7e9967e5783cc910884370. cited By 4.

[60] V.D. MALYUGIN. Arithmetical representation of petri nets. AUTOMATION AND REMOTE CONTROL, 48(5):696–703, 1987. cited By 4.

[61] R.P. Kaushal, N. Chammas, and H. Singh. A new formulation for state equation representation for petri nets. Microelectronics Reliability, 32 (8):1083–1090, 1992. doi: 10.1016/0026-2714(92)90029-K. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0026908560&partnerID=40&md5=a4da6e465cd59c38b9d32b36d2ec1c4a. cited By 0.

[62] D. Kiritsis and P. Xirouchakis. A matrix implementation of petri nets for process planning. pages 173–179, 2001.

[63] Pnml.org - pnml reference site, 2009. URL http://www.pnml.org/. [Online; accessed 21-May-2015].

[64] Iso/iec 15909-1:2004 - systems and software engineering – high-level petri nets – part 1: Concepts, definitions and graphical notation, 2004. URL http://www.iso.org/iso/catalogue_detail.htm?csnumber=43538. [Online; accessed 21-May-2015].

[65] Lom Hillah, Fabrice Kordon, Laure Petrucci-Dauchy, and Nicolas Trèves. PN standardisation: A survey. In Formal Techniques for Networked and Distributed Systems - FORTE 2006, 26th IFIP WG 6.1 International Conference, Paris, France, September 26-29, 2006., pages 307–322, 2006. doi: 10.1007/11888116_23. URL http://dx.doi.org/10.1007/11888116_23.

[66] J. Billington, S. Christensen, K. Van Hee, E. Kindler, O. Kummer, L. Petrucci, R. Post, C. Stehno, and M. Weber. The petri net markup language: Concepts, technology, and tools. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2679: 483–505, 2003. URL http://www.scopus.com/inward/record.url?eid=2-s2.

0-35248854880&partnerID=40&md5=82ea4c455b0683b147cbf9f8a5b1a557. cited By 99.

[67] Iso/iec 15909-2:2011 - systems and software engineering – high-level petri nets – part 2: Transfer format, 2011. URL http://www.iso.org/iso/catalogue_detail.htm?csnumber=43538. [Online; accessed 21-May-2015].

[68] E. Kindler. The epnk: An extensible petri net tool for pnml. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6709 LNCS:318–327, 2011. doi: 10.1007/978-3-642-21834-7_18. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-79960086227&partnerID=40&md5=695ea1e592fda71522303206030789a8. cited By 3.

[69] L.-M. Hillah, F. Kordon, C. Lakos, and L. Petrucci. Extending pnml scope: A framework to combine petri nets types. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7400 LNCS:46–70, 2012. doi: 10.1007/978-3-642-35179-2_3. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84870041075&partnerID=40&md5=5ef5e22c0a2490324ffa2441bfc1a406. cited By 2.

[70] F. Moutinho, L. Gomes, F. Ramalho, J. Figueiredo, J.P. Barros, P. Barbosa, R. Pais, and A. Costa. Ecore representation for extending pnml for input-output place-transition nets. pages 2156–2161, 2010. doi: 10.1109/IECON.2010.5675332. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-78751519789&partnerID=40&md5=b7cfa0c7f76e54f4745f0ccc3b7eb75f. cited By 0.

[71] J. Ribeiro, F. Moutinho, F. Pereira, J.P. Barros, and L. Gomes. An ecore based petri net type definition for pnml iopt models. pages 777–782, 2011. doi: 10.1109/INDIN.2011.6034992. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-80054991139&partnerID=40&md5=fc30757e8f59032f02b4d33d126353d5. cited By 3.

[72] C. Mahulea, J. Júlvez, C.R. Vázquez, and M. Silva. Continuous petri nets: Observability and diagnosis. Lecture Notes in Control and Information Sciences, 433:387–406, 2013. doi: 10.1007/978-1-4471-4276-8-19. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84866883572&partnerID=40&md5=d1221b3ab121d10302d03623f28a1b7e. cited By 0.

[73] A. Saboori and C.N. Hadjicostis. Opacity verification in stochastic discrete event systems. pages 6759–6764, 2010. doi: 10.1109/CDC.

2010.5717580. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-79953130702&partnerID=40&md5=c9b42008a9043c90839633d8aec5ca08. cited By 4.

[74] S. Velilla and M. Silva. The spy: A mechanism for safe implementation of highly concurrent systems. Annual Review in Automatic Programming, 14 (PART 1):75–81, 1988. doi: 10.1016/0066-4138(88)90012-2. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0024187397&partnerID=40&md5=940ad607e1f699f6055a587f55b7bf66. cited By 1.

[75] Xml encryption syntax and processing version 1.1, 2013. URL http://www.w3.org/TR/xmlenc-core1/. [Online; accessed 20-Jun-2015].

[76] Xml signature syntax and processing version 1.1, 2013. URL http://www.w3.org/TR/xmldsig-core1/. [Online; accessed 20-Jun-2015].

[77] Extensible markup language (xml) 1.1, 2004. URL http://www.w3.org/TR/2004/REC-xml11-20040204/#dtd. [Online; accessed 15-Jun-2015].

[78] Xml schema part 1: Structures second edition, 2004. URL http://www.w3.org/TR/xmlschema-1/. [Online; accessed 15-Jun-2015].

[79] Xml schema part 2: Datatypes second edition, 2004. URL http://www.w3.org/TR/xmlschema-1/. [Online; accessed 15-Jun-2015].

[80] Relax ng home page, 2014. URL http://relaxng.org/. [Online; accessed 15-Jun-2015].

[81] Xml path language (xpath), 1999. URL http://www.w3.org/TR/xpath/. [Online; accessed 20-Jun-2015].

[82] Xml-signature xpath filter 2.0, 2002. URL http://www.w3.org/TR/xmldsig-filter2/. [Online; accessed 20-Jun-2015].

[83] Decryption transform for xml signature, 2002. URL http://www.w3.org/TR/xmlenc-decrypt. [Online; accessed 25-Jun-2015].