THE PNML_DISC FILE FORMAT

A Petri net is a structure composed by 3 types of objects: places, transitions and arcs combined together to create a graph. The standard format used in literature to describe a Petri net is called *PNML*.
In this section the main features of PNML, a list of the advantages and the limitations, and a series of extensions to the format needed to support a wider range of tools and algorithms about Petri nets are presented. The result of these extensions is called *Pnml_DISC*.

1.1 Petri Net Markup Language

Currently in literature there are many different ASCII text based languages which are able to describe a structure or a system through the use of customized tags. One of those languages is the xml (extensible markup language). Xml is actually perfect to describe a Petri net, so from this assumption arises the idea of creating an xml-like language specifically suited to describe a Petri net. This language is called Petri Net Markup Language (PNML).

1.1.1 State of the Art

The Petri Net Markup Language (*PNML)* [1] is an international standard (ISO/IEC 15909) based on xml aimed at describing a Petri net.
The first part of the standard defines the semantic model of Petri nets and defines the following type of nets:

- high level,
- symmetric,
- place/transition.

The second part defines the syntax of the type of net defined in the first part and defines the *PNML itself* concretely.
This standard differentiates between general features typical of all types of Petri nets and specific features typical of a particular type of net. This is defined in a file called Petri Net Type Definition (PNTD) which comes always together with the main PNML file for each type of Petri net.
The PNML uses the xml syntax and follows the same schema, for this reason it is considered an extension of the xml standard format. The part 1 of the PNML standard ISO was published on 2008-03-18 and so the current version of the grammar is the result of the work of many meetings and continuous improvements in these years.

1.1.2 The Petri Net Type Definition

The Petri Net Type Definition is the PNML equivalent of the xml DTD (Document Type Definition). Since it contains the rules to define the tags and the information allowed, it actually defines the structure and the syntax of the associated PNML document.
The PNTD file contains the following information:

1. type of document (version, date, representative, etc);
2. tag of document PNML;
3. revisions of the document.

Based on the *pntd*, a document called *PetriNetTypeDefinition_DISC.pntd* is always coupled with the *Pnml_DISC*. This document, based on the standard *pntd*, contains all the information about the Pnml_DISC. In addition, it defines the range of validity of several tags, in particular it defines Pnml_DISC extension tags.
The *PetriNetTypeDefinition_DISC.pntd file* is unique and defines the schema of the Pnml_DISC.
The use of the *pntd* is optional, but it is very important to supply it together with the Pnml_DISC because it can be considered a sort of guide to correctly fill-in the Pnml_DISC avoiding inconsistencies or imprecisions. In appendix an example of *PetriNetTypeDefinition_DISC.pntd* is reported.

1.2 Advantages and limitations of the PNML

The use of PNML introduces many advantages and just a few limitations if compared to the standards such as the xml. As a result the choice of using PNML as the base for describing Petri nets in the WP4 software platform instead of more known but less specific languages.
Subsequently the attention is focused on the choice between xml or PNML, toward the format that better balances advantages and limitations. After a deep analysis of the format main features, the PNML has been selected as a starting point for the Pnml_DISC format because of the possibility to be extensible, well structured yet simple to read. Last but not least the PNML is already a standard (ISO/IEC-15909) and is supported by a variety of tools on the community.

1.2.1 Advantages

Below a list of the main advantages of using *PNML* is reported.

1. It is an international standard for Petri nets (ISO/IEC-15909).

2. It uses a specific grammar selected among: grammar version 2009, version 2.0, and version 3.0.
3. It originates from talks in many meetings.
4. It has been a consolidated and recognized format for the last 10 years.
5. It is supported by many tools on the market.
6. It is flexible enough to support different type of nets and it is open to future extensions. It is a good starting point to future format enhancements.
7. It is independent from xml and from its semantic representation.
8. It is universal.
9. It is used to describe high level nets, but it describes P/T and symmetric nets as well.
10. It is extensible and open to incorporate future versions.

The last statement is particularly important because it allows the extension of the PNML and the creation of an extended format that we have called, for simplicity, the *Pnml_DISC*. This extension is fundamental in order to support different types of Petri nets such as hybrid Petri nets.

1.2.2 Limitations

In this section the limitations of the use of the *PNML* are listed:

1. For each object there are only few labels available (e.g. place name, place marking, etc.).
2. Between the supported types of Petri nets (high level, symmetric and P/T) there are syntactic differences.
3. The extensions to the standard format are possible but the resulting file is not standard-compliant anymore. As a result, the tools supporting the PNML format are not currently able to read the PNML extensions.
4. If we change the graphics coordinates of some objects in the PNML, the tools used for a graphic representation may not be able to correctly represent these variations.

1.3 Extensions to the PNML format: the Pnml_DISC

The *PNML* format has many advantages, but is only able to describe high level, P/T or symmetric Petri nets. Since the Software Platform will support different types of Petri nets, such as hybrid Petri nets, and tools for the diagnosis of distributed systems, we need to extend the standard PNML format. The result is a format

called *Pnml_DISC*. This format includes all the information needed to describe hybrid, labeled Petri nets, and distributed systems.

During the investigation, the main requirement was to enrich the format with the supplementary properties of places and transitions needed by the software platform.

The PNML format is particularly exhaustive about graphic representations of the net, there are two kinds of graphic information: one (position) that is a reference to the position of the place or transition and another one (offset) that is a reference to the position of the name or the marking which in the coordinates are in relation to coordinates of the places or of the transitions.

The format extension is particularly needed in order to support a wider variety of tools and net types that are actually supported by the software platform. Of course, the *Pnml_DISC* can have further extensions if future tools or algorithms will require it.

In particular, we want underline the importance of using the *Pnml_DISC* to describe, besides hybrid or labeled nets, a distributed system. In fact, this format contains particular information (not available in any Petri net descriptive format) defining the parameters needed for the simulations in case of distributed systems.

These parameters are added in the extended format and are used to fully characterize a distributed system.

In the next paragraphs, each object composing a Petri net (places, transitions and arcs) is described in detail, and the differences between those elements in the standard format (PNML) and in the extended format (Pnml_DISC) are highlighted.

In general, in order to avoid user mistyping, no uppercase letters are used for the Pnml_DISC tags; these are reserved for the definitions of the same tags in the *pntd* file.

For this reason, a Pnml_DISC document must be first checked before any conversion can be made in order to avoid errors in the tags. Particularly, we check that the elements in the tags are consistent with the definitions in the *pntd* document. Besides, we require that all tags are filled because these tags are the most important ones to describe a Petri net.

The mandatory tags for a place are:

- id object
- initial_marking
- type

The mandatory tags for a transition are:

- id object
- type

The mandatory tags for an arc are:

- id object
- id source

- id target
- text

If these tags are not defined, an error is raised up by the document checker and the analysis of the document will not continue.

To conclude, an important observation: the Pnml_DISC format does not modify any of the PNML original tags, it is an extension of that format (it just adds more tags).

1.3.1 The Place Object

A place is graphically represented by a circle and in the Petri net it represents the state of an object, a system, or a user.

A place has the following descriptive tags in the standard PNML format:

- id → identifier of the object place. This tag is mandatory;
- name → contains the place text and its coordinates;
- initial_marking → initial marking of the place. This tag is mandatory;
- graphics → coordinates of graphics description of the place.

The following tags extend the standard PNML and denote the Pnml_DISC:

- type → type of the place (continuous or discrete). This tag is obligatory;
- observable → observable or not observable place;
- controllable → controllable or not controllable place;
- site_observability → site from which the place is observable.
- site_controllability → site from which the place is controllable.
- time_distribution → specifies the timing structure of discrete place;
- time_parameters → discrete place temporization parameters (temporization delay);

The meaning of the new elements is very important and is detailed below:

- **type** defines the type of place (continuous or discrete places).
  This element is important when the Petri net is hybrid: this means that there is a mix of continuous and discrete places in the same net. In *Pnml_DISC* this element is a character: C (continuous) for continuous places or D (discrete) for discrete places.
- **observable** defines the observable property of the place. It is very important to classify the object, for example in case of closed loop analysis of Petri nets. In *Pnml_DISC* it is an integer number: 0 (not observable), 1 (observable).

- **controllable** defines the controllable property of the place. It is very important to classify the object, for example in the case of closed loop analysis of Petri nets. In *Pnml_DISC* is an integer number: 0 (not controllable), 1 (controllable).
- **site_observability** defines the site from which the place is observable. It is important in the case of analysis of fault diagnosis. In the *Pnml_DISC* this element is the combination of an alphabetic character(s) representing the "site" and a numerical character representing the number of the site. Ex. s3 indicates that the place is observed by site 3.
- **site_controllability** defines the site from which the place is controllable. It is important in the case of analysis of fault diagnosis. In the *Pnml_DISC* this element is the combination of an alphabetic character(s) representing the "site" and a numerical character representing the number of the site. Ex. s2 indicates that the place is controlled from site 2.
- **time_distribution** specifies the timing structure of a discrete place. It is specified for discrete place only. It is an important element for the simulation of timed Petri nets. The Petri Nets considered have the places with a deterministic distribution, then time_distribution is equal to “det” (of default). In the *Pntd_DISC* is a sequence of 3 alphabetic characters that indicates a number between 1 and 19.
- **time_parameters** indicates the discrete place temporization parameters (temporization delay). It indicates the value to be associated with time_distribution. Time_parameter is equal to “0” (of default). It is important for the simulation of a timed Petri net. In the *Pntd_DISC* is represented by a real number.

To conclude, an extract of description of the object place in the *Pnml_DISC* format is reported below:

```
<place id="">
            <name>
               <graphics><offset x="" y=""/></graphics>
                 <text></text>
            </name>
            <initial_marking>
               <graphics><offset x="" y=""/></graphics>
               <text></text>
            </initial_marking>
            <graphics><position x="" y=""/></graphics>
            <type></type>
            <observable></observable>
```

```
        <controllable></controllable>
        <site_observability></site_observability>
        <site_controllability></site_controllability>
        <time_distribution></time_distribution>
        <time_parameters></time_parameters>
    </place>
```

1.3.2 The Transition Object

In a Petri net a transition is graphically represented by a bar (rectangle) and represents the happening event. A transition has the following descriptive tags in the standard PNML format:

- id → identifier of the object transition; this tag is obligatory;
- name → contains the text related to the transition and its graphics coordinates;
- graphics → coordinates of graphic description of the bar.

The following tags extend the standard PNML and denote the Pnml_DISC:

- type → type of transition (continuous or discrete). This tag is obligatory;
- observable → observable or not observable transition;
- controllable → controllable or not controllable transition;
- site_observability → site from which the transition is observable;
- site_controllability → site from which the transition is controllable;
- site_membership → site to which the transition belongs;
- firing_speed_min → minimum firing speed of the continuous transition;
- firing_speed_max → maximum firing speed of the continuous transition;
- time_distribution → specifies the timing structure of discrete transition;
- time_parameters → discrete transition temporization parameters (temporization delay);
- number_of_servers → the number of servers associated to discrete transition;
- priority → specifies the conflict resolution policy among discrete transitions;
- fault → the belonging fault class;
- label → label linked to the transition;
- memory → used memory policy.

Let us describe in detail the meaning of the new elements:

- **type** specifies if a transition is continuous or discrete . This element is important in the case of use a hybrid Petri net, because it is a mixture of continuous and discrete transitions. In the *Pntd_DISC* this

element is a character: C (continuous) for continuous transitions o D (discrete) for discrete transitions.

- **observable** specifies the observability property of the transitions. It is important to classify the object, for example in case of closed loop analysis of Petri nets. In the *Pntd_DISC* is an integer number: if the transition is observable (1) or if the transition is not observable (0).
- **controllable** specifies the controllable property of the transition. It is important to classify the object, for example in case of closed loop analysis of Petri nets. In the *Pntd_DISC* it is an integer number: if the transition is controllable (1) or if the transition is not controllable (0).
- **site_observability** specifies site from which the transition can be observed. It is important in the case of analysis of fault diagnosis. In the *Pnml_DISC* this element is the combination of an alphabetic character(s) to denote the "site" and a numerical character to show the number of the site. As an example s3 indicates that the transition is seen from site 3.
- **site_controllability** defines the site from which the transition is controllable. It is important in the case of analysis of fault diagnosis. In the *Pnml_DISC* this element is a combination of an alphabetic character(s) referring to the "site" and a numerical character referring to the number of the site. As an example s2 indicates that the transition is controlled from site 2.
- **site_membership** specifies the site to which the transition belongs. It is important in the case of analysis of fault diagnosis. In the *Pntd_DISC* this element is a combination of an alphabetic character(s) representing the "site" and a numerical character representing the number of the site. As an example s2 indicates that the transition belongs to site 2.
- **firing_speed_min** indicates the minimum firing speed of the transition. It is specified for continuous transitions only. It is an important parameter for the simulations of hybrid Petri nets. In the *Pntd_DISC* is represented by a real number.
- **firing_speed_max** indicates the maximum firing speed of the transition. It is specified for continuous transitions only. It is an important parameter for the simulations of Petri nets. In the *Pntd_DISC* is represented by a real number.
- **time_distribution** specifies the timing structure of a discrete transition. It is specified for discrete transitions only. It is an important element for the simulation of timed Petri nets. In *Pnml_DISC* it is a string: det (deterministic), exp (esponential), uni (uniform). In the *Pntd_DISC* is a sequence of 3 alphabetic characters that indicates a number between 1 and 19.
- **time_parameters** indicates the discrete transition temporization parameters (temporization delay). It specifies the delay of the transition. It indicates the value to be associated with time_distribution. It is important for the simulation of a timed Petri net. Time_parameters is represented by a list of parameters of arbitrary length depending on the type of distribution. The number of parameters is

equal to 1 for deterministic and exponential distribution, while for the uniform distribution is equal to 2. In the *Pntd_DISC* is represented by a list of parameters (real numbers).

- **number_of_servers** indicates the number of servers associated to a transition. It is specified for discrete transitions only. It is important for the simulation of a Petri net. In the *Pntd_DISC* is an integer number: 0 (infinite servers), k (k-servers).
- **priority** specifies the conflict resolution policy among discrete transitions. It is important for the simulation of the net because it assigns a priority to the transitions that are conflicting. In the *Pntd_DISC* is represented by an integer number.
- **fault** indicates the belonging fault class. It is important in the fault diagnosis problems. In the *Pntd_DISC* is an integer number: 0 (no fault), 1 (class of fault 1), 2 (class of fault 2), 3 (class of fault 3), and n (class of fault n).
- **label** indicates a label linked to the transition. This element can be the same for more than one transitions. It is important in case of analysis of labeled Petri nets. In the *Pntd_DISC* it is represented by a sequence of alphabetic characters.
- **memory** specifies the used memory policy for the firing transitions. It is important for the simulations of the timed Petri nets. In the *Pntd_DISC* it is represented by an alphabetic character: *e* (enabling memory), *t* (total memory).

To conclude, an extract of description of the object transition in the *Pnml_DISC* format is reported below:

```
<transition id="">
     <name>
          <graphics><offset x="" y=""/></graphics>
          <text></text>
     </name>
     <graphics><position x="" y=""/></graphics>
     <type></type>
     <observable></observable>
     <controllable></controllable>
     <site_observability></site_observability>
     <site_controllability></site_controllability>
     <site_membership></site_membership>
     <firing_speed_min></firing_speed_min>
     <firing_speed_max></firing_speed_max>
```

```
      <time_distribution></time_distribution>
      <time_parameters></time_parameters>
      <number_of_servents></number_of_servents>
      <priority></priority>
      <fault></fault>
      <label></label>
      <memory></memory>
</transition>
```

An arc is an arrow that starts from a place or a transition and arrives to a transition or a place. In a Petri net it represents the link between the places and the transitions.
An arc, is denoted, in a PNML file, by the following tags:

- id → identifier of the object arc. This tag is obligatory;
- source → object id of the arc start point. This tag is obligatory;
- target → object id of the arc end point. This tag is obligatory;
- text → weight of the arc. This tag is obligatory;
- graphics → coordinates of graphic description of the arrow.

The above tags are defined in the PNML format and there is no need to extend this format using more tags because they describe exhaustively the information of an arc object.
In conclusion the object arc is described in the Pnml_DISC using the same tags of the standard PNML. Below there is a brief description of these tags:

- **source** indicates the object the arrow starts from. This field contains the id of the start point object and not the name of the object.
- **target** indicates the object the arrow arrives to. This field contains the id of the end point object and not the name of the object.
- **text** specifies the weight of the arc, namely the quantity of tokens transported through the arc.

To conclude, an extract of description of the object arc in the *Pnml_DISC* format is reported below:

```
<arc id="" source="" target="">
            <inscription>
               <graphics><offset x="" y="" /></graphics>
               <text></text>
```

```
   </inscription>
   <graphics><position x="" y="" /></graphics>
</arc>
```

1.4 A PNML_DISC example

Below, the PNML_DISC description of the Petri net reported in Figure 1 is represented:

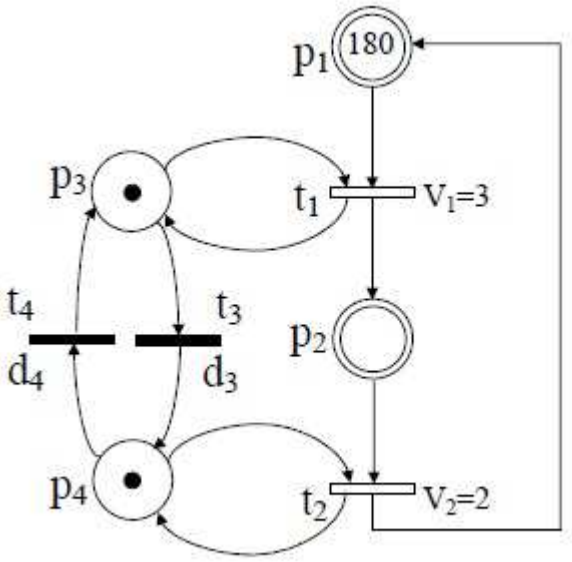

Figure 1 - A hybrid Petri net example

```
<pnml xmlns="http://www.informatik.hu-berlin.de/top/pnml/ptNetb">
   <net id="n1" type="http://www.informatik.hu-berlin.de/top/pntd/ptNetb">
       <name>
           <text>rete7_matlab_disc.m</text>
       </name>
       <page id="page1">
         <place id="p1">
            <name>
               <graphics><offset x="0" y="0"/></graphics>
               <text>P1</text>
            </name>
            <initial_marking>
               <graphics><offset x="0" y="0"/></graphics>
               <text>180</text>
            </initial_marking>
            <graphics><position x="10" y="10"/></graphics>
```

```
   <type>C</type>
   <observable></observable>
   <controllable></controllable>
   <site_observability></site_observability>
   <site_controllability></site_controllability>
   <time_distribution></time_distribution>
   <time_parameters></time_parameters>
</place>
<place id="p2">
   <name>
      <graphics><offset x="0" y="0"/></graphics>
      <text>P2</text>
   </name>
   <initial_marking>
      <graphics><offset x="0" y="0"/></graphics>
      <text>0</text>
   </initial_marking>
   <graphics><position x="20" y="20"/></graphics>
   <type>C</type>
   <observable></observable>
   <controllable></controllable>
   <site_observability></site_observability>
   <site_controllability></site_controllability>
   <time_distribution></time_distribution>
   <time_parameters></time_parameters>
</place>
<place id="p3">
   <name>
      <graphics><offset x="0" y="0"/></graphics>
      <text>P3</text>
   </name>
   <initial_marking>
      <graphics><offset x="0" y="0"/></graphics>
      <text>1</text>
   </initial_marking>
   <graphics><position x="31" y="31"/></graphics>
   <type>D</type>
   <observable></observable>
   <controllable></controllable>
   <site_observability></site_observability>
```

```
   <site_controllability></site_controllability>
   <time_distribution>det</time_distribution>
   <time_parameters>0</time_parameters>
</place>
<place id="p4">
   <name>
      <graphics><offset x="0" y="0"/></graphics>
      <text>P4</text>
   </name>
   <initial_marking>
      <graphics><offset x="0" y="0"/></graphics>
      <text>1</text>
   </initial_marking>
   <graphics><position x="43" y="43"/></graphics>
   <type>D</type>
   <observable></observable>
   <controllable></controllable>
   <site_observability></site_observability>
   <site_controllability></site_controllability>
   <time_distribution>det</time_distribution>
   <time_parameters>0</time_parameters>
</place>
<transition id="t1">
   <name>
      <graphics><offset x="0" y="0"/></graphics>
      <text>T1</text>
   </name>
   <graphics><position x="10" y="10"/></graphics>
   <type>C</type>
   <observable>1</observable>
   <controllable></controllable>
   <site_observability></site_observability>
   <site_controllability></site_controllability>
   <site_membership></site_membership>
   <firing_speed_min>0</firing_speed_min>
   <firing_speed_max>3</firing_speed_max>
   <time_distribution></time_distribution>
   <time_parameters></time_parameters>
   <number_of_servers></number_of_servers>
   <priority>1</priority>
```

```
   <fault></fault>
   <label>T1</label>
   <memory></memory>
</transition>
<transition id="t2">
   <name>
      <graphics><offset x="0" y="0"/></graphics>
      <text>T2</text>
   </name>
   <graphics><position x="20" y="20"/></graphics>
   <type>C</type>
   <observable>1</observable>
   <controllable></controllable>
   <site_observability></site_observability>
   <site_controllability></site_controllability>
   <site_membership></site_membership>
   <firing_speed_min>0</firing_speed_min>
   <firing_speed_max>2</firing_speed_max>
   <time_distribution></time_distribution>
   <time_parameters></time_parameters>
   <number_of_servers></number_of_servers>
   <priority>1</priority>
   <fault></fault>
   <label>T2</label>
   <memory></memory>
</transition>
<transition id="t3">
   <name>
      <graphics><offset x="0" y="0"/></graphics>
      <text>T3</text>
   </name>
   <graphics><position x="31" y="31"/></graphics>
   <type>D</type>
   <observable>1</observable>
   <controllable></controllable>
   <site_observability></site_observability>
   <site_controllability></site_controllability>
   <site_membership></site_membership>
   <firing_speed_min></firing_speed_min>
   <firing_speed_max></firing_speed_max>
```

```
   <time_distribution>det</time_distribution>
   <time_parameters>90</time_parameters>
   <number_of_servers>0</number_of_servers>
   <priority>1</priority>
   <fault></fault>
   <label>T3</label>
   <memory></memory>
</transition>
<transition id="t4">
   <name>
      <graphics><offset x="0" y="0"/></graphics>
      <text>T4</text>
   </name>
   <graphics><position x="43" y="43"/></graphics>
   <type>D</type>
   <observable>1</observable>
   <controllable></controllable>
   <site_observability></site_observability>
   <site_controllability></site_controllability>
   <site_membership></site_membership>
   <firing_speed_min></firing_speed_min>
   <firing_speed_max></firing_speed_max>
   <time_distribution>det</time_distribution>
   <time_parameters>60</time_parameters>
   <number_of_servers>0</number_of_servers>
   <priority>1</priority>
   <fault></fault>
   <label>T4</label>
   <memory></memory>
</transition>
<arc id="a1" source="p1" target="t1">
   <inscription>
      <graphics><offset x="0" y="0" /></graphics>
      <text>1</text>
   </inscription>
   <graphics><position x="0" y="0" /></graphics>
</arc>
<arc id="a2" source="t2" target="p1">
   <inscription>
      <graphics><offset x="0" y="0" /></graphics>
```

```
      <text>1</text>
   </inscription>
   <graphics><position x="0" y="0" /></graphics>
</arc>
<arc id="a3" source="t1" target="p2">
   <inscription>
      <graphics><offset x="0" y="0" /></graphics>
      <text>1</text>
   </inscription>
   <graphics><position x="0" y="0" /></graphics>
</arc>
<arc id="a4" source="p2" target="t2">
   <inscription>
      <graphics><offset x="0" y="0" /></graphics>
      <text>1</text>
   </inscription>
   <graphics><position x="0" y="0" /></graphics>
</arc>
<arc id="a5" source="p3" target="t1">
   <inscription>
      <graphics><offset x="0" y="0" /></graphics>
      <text>1</text>
   </inscription>
   <graphics><position x="0" y="0" /></graphics>
</arc>
<arc id="a6" source="t1" target="p3">
   <inscription>
      <graphics><offset x="0" y="0" /></graphics>
      <text>1</text>
   </inscription>
   <graphics><position x="0" y="0" /></graphics>
</arc>
<arc id="a7" source="p3" target="t3">
   <inscription>
      <graphics><offset x="0" y="0" /></graphics>
      <text>1</text>
   </inscription>
   <graphics><position x="0" y="0" /></graphics>
</arc>
<arc id="a8" source="t4" target="p3">
```

```
            <inscription>
               <graphics><offset x="0" y="0" /></graphics>
               <text>1</text>
            </inscription>
            <graphics><position x="0" y="0" /></graphics>
         </arc>
         <arc id="a9" source="p4" target="t2">
            <inscription>
               <graphics><offset x="0" y="0" /></graphics>
               <text>1</text>
            </inscription>
            <graphics><position x="0" y="0" /></graphics>
         </arc>
         <arc id="a10" source="t2" target="p4">
            <inscription>
               <graphics><offset x="0" y="0" /></graphics>
               <text>1</text>
            </inscription>
            <graphics><position x="0" y="0" /></graphics>
         </arc>
         <arc id="a11" source="t3" target="p4">
            <inscription>
               <graphics><offset x="0" y="0" /></graphics>
               <text>1</text>
            </inscription>
            <graphics><position x="0" y="0" /></graphics>
         </arc>
         <arc id="a12" source="p4" target="t4">
            <inscription>
               <graphics><offset x="0" y="0" /></graphics>
               <text>1</text>
            </inscription>
            <graphics><position x="0" y="0" /></graphics>
         </arc>
      </page>
   </net>
</pnml>
```

1.5 A Pntd_DISC example

```
<?xml version="1.0">
<!-- This is the version 1.0 of pntd document -->

<grammar ns="http://www.informatik.hu-berlin.de/top/pnml/ptNetb"
      xmlns="http://relaxng.org/ns/structure/1.0"
      xmlns:a="http://relaxng.org/ns/compatibility/
      annotations/1.0">
<a:documentation>
      Petri Net Type Definition for Place/Transition
      nets (bases on basic PNML)
      RELAX NG implementation of demo1.pntd
      version: 1.0
      2009
      Antonio Solinas antonio.solinas@akhela.com
</a:documentation>
<a:documentation>
      In this document we define the range of validity of
      each parameter that is used in the PNML format.
</a:documentation>
<a:documentation>
      We extend the PNML format because in the standard format
      there are not the parameters to define the hybrid Petri
      net.
</a:documentation>

<define name="net.labels" combine="interleave">
<a:documentation>
      A P/T net may have a name.
</a:documentation>
<interleave>
<optional><ref name="Name"/></optional>
</interleave>
</define>
<define name="place.labels" combine="interleave">
<a:documentation>
      A place of a P/T net may have a name, an initial
      marking, a type, a observable, a controllable,
```

```
      a site_observability, a site_controllability,
      a time_distribution and a time_parameters.
      </a:documentation>
      <interleave>
      <optional><ref name="name"/></optional>
      <optional><ref name="initial_marking"/></optional>
      <optional><ref name="type"/></optional>
      <optional><ref name="observable"/></optional>
      <optional><ref name="controllable"/></optional>
      <optional><ref name="site_observability"/></optional>
      <optional><ref name="site_controllability"/></optional>
      <optional><ref name="time_distribution"/></optional>
      <optional><ref name="time_parameters"/></optional>
      </interleave>
      </define>
<a:documentation>
      The name is a combination of numeric and/or literal
      characters.
      The initial marking is a value; it is a integer number
      for discrete place or real number for continuous place.
      The type is a single character: D for discrete place
      or C for continuous place.
      The observable is a integer number: 0 if it is not
      observable, 1 if it is observable.
      The controllable is a integer number: 0 if it is not
      controllable, 1 if it is controllable.
      The site_observability is a single character (s) and
      an integer number.
      The site_controllability is a single character (s)
      and an integer number.
      The time_distribution is a sequence of 3
      characters to express the number since 1 to 19.
      The time_parameters is a real number.
</a:documentation>
<define name="transition.labels" combine="interleave">
<a:documentation>
      A transition of a P/T net may have a name, a type,
      an observable, a controllable, a site_observability,
      a site_controllability, a site_membership, a firing_speed_min,
      a firing_speed_max, a time_distribution,
```

```
      a time_parameters, a number_of_servers, a priority, a fault,
      a label and a memory.
</a:documentation>
<interleave>
      <optional><ref name="name"/></optional>
      <optional><ref name="type"/></optional>
      <optional><ref name="observable"/></optional>
      <optional><ref name="controllable"/></optional>
      <optional><ref name="site_observability"/></optional>
      <optional><ref name="site_controllability"/></optional>
      <optional><ref name="site_membership"/></optional>
      <optional><ref name="firing_speed_min"/></optional>
      <optional><ref name="firing_speed_max"/></optional>
      <optional><ref name="time_distribution"/></optional>
      <optional><ref name="time_parameters"/></optional>
      <optional><ref name="number_of_servers"/></optional>
      <optional><ref name="priority"/></optional>
      <optional><ref name="fault"/></optional>
      <optional><ref name="label"/></optional>
      <optional><ref name="memory"/></optional>
</interleave>
</define>
<a:documentation>
      The name is a combination of numeric and/or literal
      characters.
      The type is a single character: D for discrete
      transition or C for continuous transition.
      The observable is a integer number: 0 if it is not
      observable, 1 if it is observable.
      The controllable is a integer number: 0 if it is
      not controllable, 1 if it is controllable.
      The site_observability is a single character (s)
      and a integer number.
      The site_controllability is a single character
      (s) and a integer number.
      The site_membership is a single character (s)
      and a integer number.
      The firing_speed_min is a real number.
      The firing_speed_max is a real number.
      The time_distribution is a sequence of 3
```

```
      characters to express the number since 1 to 19.
      The time_parameters is a list of parameters.
      The number_of_servers is a integer number:
      0 for infinite servers, k for k-servers.
      The priority is a integer number.
      The fault is a integer number: 0 not fault,
      the next numbers indicates the class of fault.
      The label is a sequence of literal characters.
      The memory is a single character:
      e (enable) or t (total).
</a:documentation>
<define name="arc.labels" combine="interleave">
<a:documentation>
      An arc of a P/T net may have an inscription.
</a:documentation>
<interleave>
<optional><ref name="PTArcInscription"/></optional>
</interleave>
</define>
<a:documentation>
      The inscription is the weight of the arc.
      It is a real number.
</a:documentation>
</grammar>

<!-- BEGIN RCS-Info
Revision 1.1 2009/12/15 15:12:23 Antonio
Initial revision
A.1. PETRI_NET_TYPE_DEFINITION_DISC 33
Revision 1.2 2009/12/21 15:14:10 Antonio
Insert new label
Revision 1.3 2010/01/13 10:07:18 Antonio
Correct characters of the text
Revision 1.4 2010/02/17 11:03:49 Antonio
Correct order of the elements
Revision 1.5 2010/02/22 15:52:13 Antonio
Added site_controllability
END RCS-Info
-->
<!-- Table to time distribution
```

```
Deterministic or Det:1,
Exponential or Exp:2,
Uniform or Uni:3,
Poisson or Poi:4,
Rayleigh or Ray:5,
Weibull or Wbl:6,
Beta or Bet:7,
Chisquare2 or Chi:8,
Extreme Value or Eva:9,
F or FFF:10,
Gamma or gam:11,
Generalized Extreme Value or gev:12,
Generalized Pareto or gpo:13,
Lognormal or log:14,
Noncentral F or ncf:15,
Noncentral t or nct:16,
Noncentral Chi-square or ncx:17,
Normal or nor:18,
T or ttt:19. -->
<!-- The number that defines some parameters are all non negative numbers -->
```

BIBLIOGRAPHY

[1] **http://www.pnml.org/index.php, Petri Net Markup Language - Home Page.** [Online]