A primer on the Petri Net Markup Language and ISO/IEC 15909-2

Lom Hillah$^1$ Ekkart Kindler$^2$ Fabrice Kordon$^1$
Laure Petrucci$^3$ Nicolas Trèves$^4$

$^1$LIP6, Université Pierre et Marie Curie, Paris, France
$^2$Technical University of Denmark (DTU), Lyngby, Denmark
$^3$LIPN, University Paris 13, Villetaneuse, France
$^4$CEDRIC, CNAM, Paris, France
Genesis of ISO/IEC 15909

- formal definition of high-level and place/transition nets
- International Standard since December 2004
- an amendment defining symmetric nets is in final stages of discussion

- aims at defining an exchange format, PNML
- is accompanied by tool support
- should become an international standard by mid-2010

**Part 3 (2009–2013?)**
- concerned with extensions of nets
  - how to define new types of nets, nodes, arcs
  - include structuring mechanisms
  - extension of the PNML exchange format
# Genesis of ISO/IEC 15909

- **formal definition** of high-level and place/transition nets
- **International Standard** since December 2004
- an amendment defining **symmetric nets** is in final stages of discussion

- aims at defining an exchange format, PNML
- is accompanied by **tool support**
- should become an international standard by mid-2010

## Part 3 (2009–2013?)
- concerned with **extensions** of nets
  - how to define **new types** of nets, nodes, arcs
  - include **structuring mechanisms**
  - extension of the PNML exchange format
Genesis of ISO/IEC 15909

- **formal definition** of high-level and place/transition nets
- **International Standard** since December 2004
- an amendment defining **symmetric nets** is in final stages of discussion

- aims at defining an **exchange format**, PNML
- is accompanied by **tool support**
- should become an international standard by mid-2010

Part 3 (2009–2013?)
- concerned with **extensions** of nets
  - how to define **new types** of nets, nodes, arcs
  - include **structuring mechanisms**
  - extension of the PNML exchange format
# Genesis of ISO/IEC 15909

- **formal definition** of high-level and place/transition nets
- **International Standard** since December 2004
- an amendment defining **symmetric nets** is in final stages of discussion

- aims at defining an **exchange format**, PNML
- is accompanied by **tool support**
- should become an international standard by mid-2010

## Part 3 (2009–2013?)
- concerned with **extensions** of nets
  - how to define **new types** of nets, nodes, arcs
  - include **structuring mechanisms**
  - extension of the **PNML** exchange format
### Motivation

#### Aims

1. Easily and unambiguously exchange models between tools
2. Establish semantical correspondence among Petri net types

#### Key issues

1. Choice of exchange technology (aim 1)
2. Support tool specific information (aim 1)
3. Share common concepts (aim 2)
4. Manage extensions and variants (aim 2)
Motivation

Aims

1. easily and unambiguously exchange models between tools
2. establish semantical correspondence among Petri net types

Key issues

1. choice of exchange technology (aim 1)
2. support tool specific information (aim 1)
3. share common concepts (aim 2)
4. manage extensions and variants (aim 2)
Outline

1. Choices
2. Meta-model hierarchy and essential ingredients
3. Petri net type meta-models
4. PNML syntax
5. Conclusion & Perspectives
## Choices & Underlying Technologies

### Key issue 1: exchange technology
- **XML technology** (static definition of concepts: XML-Schema, Relax-NG, etc.)

### Key issues 2 and 4: tool specific information, extensions and variants
- Key issue 4 refines key issue 2: it deals with **semantics**
- Requires **structured meta-models** to handle these extensions

### Key issue 3: common concepts
- Petri net type = assembly of elementary concepts
  - TPN = PTN + time, FIFO-Nets = PTN + queues, etc.
- Requires **advanced structuring of meta-models** to handle these compositions
Choices & Underlying Technologies

Key issue 1: exchange technology
- **XML technology** (static definition of concepts: XML-Schema, Relax-NG, etc.)

Key issues 2 and 4: tool specific information, extensions and variants
- key issue 4 refines key issue 2: it deals with **semantics**
- requires **structured meta-models** to handle these extensions

Key issue 3: common concepts
- Petri net type = assembly of elementary concepts
  - TPN = PTN + time, FIFO-Nets = PTN + queues, etc.
- requires **advanced structuring of meta-models** to handle these compositions
Choices & Underlying Technologies

Key issue 1: exchange technology
- **XML technology** (static definition of concepts: XML-Schema, Relax-NG, etc.)

Key issues 2 and 4: tool specific information, extensions and variants
- key issue 4 refines key issue 2: it deals with **semantics**
- requires **structured meta-models** to handle these extensions

Key issue 3: common concepts
- Petri net type = assembly of elementary concepts
  - TPN = PTN + time, FIFO-Nets = PTN + queues, etc.
- requires **advanced structuring of meta-models** to handle these compositions
Choices & Underlying Technologies

Key issue 1: exchange technology
- XML technology (static definition of concepts: XML-Schema, Relax-NG, etc.)

Key issues 2 and 4: tool specific information, extensions and variants
- Key issue 4 refines key issue 2: it deals with semantics
- requires structured meta-models to handle these extensions

Key issue 3: common concepts
- Petri net type = assembly of elementary concepts
  TPN = PTN + time, FIFO-Nets = PTN + queues, etc.
- requires advanced structuring of meta-models to handle these compositions

Model-Driven Engineering techniques
Meta-model Hierarchy and Essential Ingredients

- algebra extensions
- definition of labels
- definition of algebras
- pages, nodes, arcs
- no labels but names
- basic graphics

PNML Core Model
Meta-model Hierarchy and Essential Ingredients

- PT-Net
- SymmetricNet
- PNML Core Model

- algebra extensions
- definition of labels
- definition of algebras
- pages, nodes, arcs
- no labels but names
- basic graphics
Meta-model Hierarchy and Essential Ingredients

- PNML Core Model
  - pages, nodes, arcs
  - no labels but names
  - basic graphics

- PT-Net
  - algebra extensions
  - definition of labels
  - definition of algebras

- SymmetricNet

- HLPNG

Meta-model hierarchy and essential ingredients
Definition of an algebra for Symmetric nets
Definition of an algebra for Symmetric nets
Petri Net Type Meta-models — Symmetric Nets

Definition of an algebra for Symmetric nets
Petri Net Type Meta-models — Symmetric Nets

Definition of an algebra for Symmetric nets
Petri Net Type Meta-models — Symmetric Nets

Definition of an algebra for Symmetric nets

- FiniteIntRanges
- FiniteEnumerations
- CyclicEnumerations
- Partitions
- Booleans
- Multisets
- Dots
- SymmetricNets
- HLCoreStructure

<<import>>

<<merge>>
Petri net type meta-models — High-Level net graphs

Extension of the algebra for High-Level Petri Net Graphs

Terms

HLPNGs

SymmetricNets
Petri net type meta-models — High-Level net graphs

Extension of the algebra for High-Level Petri Net Graphs
Extension of the algebra for High-Level Petri Net Graphs

- Terms
- Lists
- Strings
- Integers
- ArbitraryDeclarations
- HLPNGs
- SymmetricNets
Petri net type meta-models — High-Level net graphs

Extension of the algebra for High-Level Petri Net Graphs

High-level meta-model elements and their PNML constructs

<table>
<thead>
<tr>
<th>Model element</th>
<th>PNML element</th>
<th>PNML attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terms::UserSort</td>
<td>usersort</td>
<td>declaration: IDREF</td>
</tr>
<tr>
<td>Terms::UserOperator</td>
<td>useroperator</td>
<td>declaration: IDREF</td>
</tr>
<tr>
<td>ArbitraryDeclarations::ArbitrarySort</td>
<td>arbitrarysort</td>
<td></td>
</tr>
</tbody>
</table>

...
<place id="p1">
  <graphics>
    <position x="20" y="20"/>
  </graphics>
  <name>
    <text>ready</text>
    <graphics>
      <offset x="0" y="-10"/>
    </graphics>
  </name>
  <initialMarking>
    <text>3</text>
    <toolspecific tool="org.pnml.tool" version="1.0">
      <tokengraphics>
        <tokenposition x="-2" y="-2"/>
        <tokenposition x="2" y="0"/>
        <tokenposition x="-2" y="2"/>
      </tokengraphics>
    </toolspecific>
  </initialMarking>
</place>
<place id="p1">
  
  <graphics>
    <position x="20" y="20"/>
  </graphics>

  <name>
    <text>ready</text>
  </name>

  <initialMarking>
    <text>3</text>
    <toolspecific tool="org.pnml.tool" version="1.0">
      <tokengraphics>
        <tokenposition x="-2" y="-2"/>
        <tokenposition x="2" y="0"/>
        <tokenposition x="-2" y="2"/>
      </tokengraphics>
    </toolspecific>
  </initialMarking>

</place>
<place id="p1">
  <graphics>
    <position x="20" y="20"/>
  </graphics>
  <name>
    <text>ready</text>
    <graphics>
      <offset x="0" y="-10"/>
    </graphics>
  </name>
  <initialMarking>
    <text>3</text>
    <toolspecific tool="org.pnml.tool" version="1.0">
      <tokengraphics>
        <tokenposition x="-2" y="-2" />
        <tokenposition x="2" y="0" />
        <tokenposition x="-2" y="2" />
      </tokengraphics>
    </toolspecific>
  </initialMarking>
</place>
PNML syntax

<transition id="t1">
    <graphics>
        <position x="60" y="20"/>
    </graphics>
</transition>

<arc id="a1" source="p1" target="t1">
    <graphics>
        <position x="30" y="5"/>
        <position x="60" y="5"/>
    </graphics>
    <inscription>
        <text>2</text>
        <graphics>
            <offset x="0" y="5"/>
        </graphics>
    </inscription>
</arc>
PNML syntax: an example

```
<transition id="t1">
  <graphics>
    <position x="60" y="20"/>
  </graphics>
</transition>

<arc id="a1" source="p1" target="t1">
  <graphics>
    <position x="30" y="5"/>
    <position x="60" y="5"/>
  </graphics>
  <inscription>
    <text>2</text>
    <graphics>
      <offset x="0" y="5"/>
    </graphics>
  </inscription>
</arc>
```
<transition id="t1">
  <graphics>
    <position x="60" y="20"/>
  </graphics>
</transition>

<arc id="a1" source="p1" target="t1">
  <graphics>
    <position x="30" y="5"/>
    <position x="60" y="5"/>
  </graphics>
  <inscription>
    <text>2</text>
    <graphics>
      <offset x="0" y="5"/>
    </graphics>
  </inscription>
</arc>
Conclusion

- Petri Net types tackled:
  - P/T nets (also in high-level notation)
  - Symmetric nets
  - High Level Petri net graphs

- reference implementation of APIs companion to the standard: PNML

- Framework relies on mature MDE technology: Eclipse/EMF
- automatic generation of APIs from the meta-models in the standard
- assessment of choices

- MDE techniques require enrichment-based inheritance
  - bottom-up approach of Petri net types design
Perspectives

- the standard needs to spread out
  - use by more tools
  - actualisation of current support

**Challenges raised by part 3**
- more variability in Petri net types
- guided support of hierarchy
- handling of several dimensions (time, stochastics, etc.)
  - bottom-up approach of Petri net types design really required here
The Standard NEEDS You

Les devises Shadok

S’il n’y a pas de solution c’est qu’il n’y a pas de problème.

Les devises Shadok

En essayant continuellement on finit par réussir. Donc: plus ça rate, plus on a de chances que ça marche.