The Petri Net Markup Language
theory and practice

Lom Messan Hillah
Université Pierre et Marie Curie
Outline

- Motivations
- PNML Framework: how to use it?
- Application examples (Coloane, Validation, Dot)
- How is it built? (MDE)
Outline

- Limitations
- Ideas for improvement
- Conclusion
- Resources
Exchanging PN models...

PNML is about exchanging Petri net models, not XML
Exchanging PN models...

- The XML syntax is irrelevant to design
- Tools must (automatically) deal with it
- Keep compliance with the standard
- Tool-specific tag for non-standard information
- Best effort strategy otherwise
Exchanging PN models...

Actually, what are their respective tools doing?

LOAD

<pn xmlns="......">
<net type="http://www">
<name>
  dining philosophers
</name>
<page id="page1">
<place id="p1">

SAVE

CREATE

FETCH
Motivations
PNML learning difficulties

- Developers are not yet familiar with the standard
- Conceptual part is the one to fully comprehend
- The core semantics do not lie in the XML
- The standard is not freely available
- We need, at least, an entry point
Easing the access to PNML

- Transparent, easy way to handle PNML documents
- Help developers concentrate on the core of their applications, not PNML
- Keeping up-to-date and compliant
- Keep the door open for future extensions
- Reference implementations should make it work
PNML Framework

- Aims at being a reference implementation of PNML
- It is first intended to be used as a library by tools
  - Easy to use API to handle PNML documents
- Two use cases of a PN tool for handling PNML:
  - import Petri nets from PNML documents
  - export Petri nets into PNML documents
PNML Framework at work

Petri net tool T1

Petri net tool T2

Petri net tool T3

PTML Framework in an existing tool

PTML document

PTML Framework in a new independent tool

T3's implementation of PTML

Best effort PTML handling

PNML Framework integrated in an existing tool

PNML Framework integrated in a new independent tool

PNML document

Ready

In_Race

Podium

Start_Race

End_Race

Get_prepared

Ready

Start_Race

In_Race

End_Race

Podium

Get_prepared
Integrating PNML Framework

Developer's PNML handler

Application Driver

PNML Framework

Create models

Save models

Fetch models

Save models

Populate repository

Proprietary model repository

Developer's parser

Petri net models in proprietary format

Load

Fetch

Payumoto

PNML repository

PNML exporter

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Petri net models in proprietary format

PNML repository

Save models

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

PNML repository

Create models

Fetch and check

Save

Petri net models in PNML

Proprietary model repository

Payumoto

Fetch models

Save models

Populate repository

Proprietary model repository

 PNML: Practice - PN'09
Outline

Motivations

PNML Framework

Application examples
Application examples
(Short demo)

- Coloane (cross-platform Petri net editor)
- PNML validation
- PNML to dot
- PNML to Coq
Outline

How is it built?

Motivations

PNML Framework

Application examples
How is it built?

- Model-Driven Engineering principles:
  - First model your domain-specific language (DSL)
  - Customize code generation
  - Then generate code, customize again, round-trip, ...
- Advantages: high-level, less error-prone, target language (relatively) independent, sync. model-code, maintainable, extensible
How is it built?

1. Petri net type T1
   Ecore model (EMF)

2. Petri net type T1
   genmodel (EMF)

3. Annotated model for
   code generation

4. Code generation templates
   (configuration for PNML)

5. PNML Export API

   /** Export to PNML */
   public String toPNML();
   .......

6. PNML Import API

   /** Import from PNML */
   public pnModel
   fromPNML();
   .......

7. JET Code Generator

   Code generation

8. Metamodels for
   Petri net types

9. CODE
   GENERATOR

10. Standard version

11. CREAT
12. ESAVE
13. LOAD
14. FETCH

PNML: Practice - PN’09
Architecture

PNML: Practice - PN'09
How to support PNML?

- Use PNML Framework: a few days
  - we have already implemented the trickiest part
  - easy to use high-level API, documented

- Build your own tool from scratch: (?)
  - Dig into the standard: metamodels, OCL constraints, PNML grammar...
Petri net tools market (PNDB)

- 72 tools (June 2009)
  - ≈ 33% in Java (24)
  - ≈ 17% in C++ (12)
  - ≈ 26% non explicitly declared (19)
  - ≈ 23% (17) in Ada, C, Python, Smalltalk, Tcl/Tk, ...

How many are discontinued?
Limitations

- Static code generation API, w.r.t. the metamodels

- Every metamodel update implies a code regeneration

- No native front-end, i.e., editor (but a lot of them out there + new initiatives)

- Java
Ideas for improvement

- Dynamic metamodel plug-in mechanism
- Grammar generation for new Petri net types
- New target languages
PNML: theory and practice

- Interoperability of Petri net tools
- Designed for extension
- PNML Framework: a reference implementation for PN tools to use to handle PNML documents
- Easy to use, fast integration
- New applications are welcome
- New ideas are welcome
Web-based repository for PN models in PNML

Freely accessible: you may download or propose models

Free and immediate PNML document validation

RESTful: you can interact with it from your own client application (URI-based invocations)
Webography

http://www.pnml.org


http://www.pnmlweb.org (Fall 2009)
Social process around Petri net types
Social process around PN types

- ISO/IEC 15909 defines PN types, implemented by PNML, which is open
- But the standard does not provide all PN types
- Community should have the ability to propose and push for new features and types to be standardized
- Set up a common informal process to do so
Social process around PN types

- What? new PN features and types
- Where? interactive web platform: “forum”...
- When? anytime
- How? votes, conclusive experiments on sound proposals