



Instructions for the **CALL FOR MODEL** OF THE **MODEL CHECKING CONTEST @ PETRI NETS** <http://mcc.lip6.fr>

Thanks for considering to submit one or several models to the Model Checking Contest. This README explains the procedure to carry out in order to successfully submit your model(s) to the MCC.

Step 1: Filling the Model Form

Three steps are required to build the model form:

1. Provide the description of each model you want to propose by filling the file **template/DescModel.tex** in, one file per model. Examples of model description from the previous editions of MCC are provided for your information in the **examples** folder. In case you have several models, organize them into their respective folder, like the provided examples. To do so, duplicate the template folder and rename it.
2. Once a model description is provided, compile the model document with **pdflatex**. The main file (**main.tex** does not need to be changed, you only have to fill the macros located in the **DescModel.tex** file).
3. Check the content you provided appears properly in the generated PDF. If you are satisfied with the result, then remove the compilation artifacts other than the generated PDF.

For any help regarding this step, contact Fabrice.Kordon@lip6.fr.

Step 2: Producing the PNML Files

Each instance of your model must have a corresponding PNML file. Place the PNML files in Place/Transition net in the PNML/PT folder, and the PNML files in Symmetric net in the PNML/COLORED folder. The following help is provided to you:

- The examples provided in the Model Submission Kit also come with their **PNML** files. You may use them as reference to get started in producing your PNML files.
- Alternatively, an Eclipse project archive of programs to produce some models using the **PNML Framework API** is available at <http://pnml.lip6.fr/pnmlframework/mcc/MCC4PNMLLIP6FR.zip>. PNML Framework is the companion tool to the PNML Standard. It provides a reference implementation of the PNML standard in Eclipse Modeling Framework (EMF), thus relying on Model-Driven Engineering techniques.

Since these example programs show how to create models using the PNML Framework API, they are mainly appropriate in a context where you are working in a Eclipse/Java environment, and you are able to **download and use the PNML Framework library**.

If you use **Coloane** to draw your Petri net models, then:

- For P/T nets, export your models into the CAMI format (**Right-click on the model** → **Export...** → **Coloane** → **CAMI File**) and use the Cami2Pnml model converter to export the CAMI model into PNML. The Cami2Pnml tool is available at <http://pnml.lip6.fr/camipnml/introduction.html>.
- For Symmetric nets, export your models into the GrML format (**Right-click on the model** → **Export..** → **Coloane** → **GrML File**) and use the GrML2Pnml model converter to export the GrML model in to PNML. The GrML2Pnml tool is available at <http://pnml.lip6.fr/applications.html#grmlpnml>

Another way to start producing PNML is to generate your own parser, using generators like:

- **JAXB** (that generates a Java API out of XML Schema Definitions. Also see: <http://www.oracle.com/technetwork/articles/javase/index-140168.html>)
- **Apache XMLBeans**, that also provides XML Schema to Java data binding.
- **Code Synthesis**, a XML Schema to C++ data binding compiler.
- For Python, please refer to the following page: <https://wiki.python.org/moin/PythonXml>

The above tools rely on XML Schema Definitions (XSD), while **PNML grammar** is in **RELAX NG**. To convert RELAX NG into XSD, you may use **Trang**, a multi-format schema converter based on RELAX NG. For a list of other softwares in the RELAX NG ecosystem, refer to the following page: <http://relaxng.org/#software>.

Finally, you may also directly use any tool of your own, possibly by relying on XML libraries in your favorite language (e.g. see: <http://xmlsoft.org>).

For any help regarding this step, contact Lom-Messan.Hillah@lip6.fr.

Model Submission

After having successfully completed all of the above steps, and successfully checked the output for every model you want to submit:

1. Rename this top-level folder into MCC-ModelSubmissionKit-YourName, (replace YourName with your actual LastName.FirstName);
2. Zip that folder and send it to Fabrice.Kordon@lip6.fr.

Thanks for your contribution to the MCC!