

TP 01: Creation of a modeling language for Petri Nets (PNs)

The objective of this practical work is the development of a graphical tool for the creation of Petri Nets. This practical work contains three steps:

- 1-Definition of the abstract syntax of PN
- 2- Definition of the concrete syntax of the PN
- 3- Creation and use of a PN model.

But before you start you need to download and install three applications

- Python: version greater than 3
- node.js
- ATOMPM

Now we can start, the following figure represents the start page for creating a new account if you do not have one.

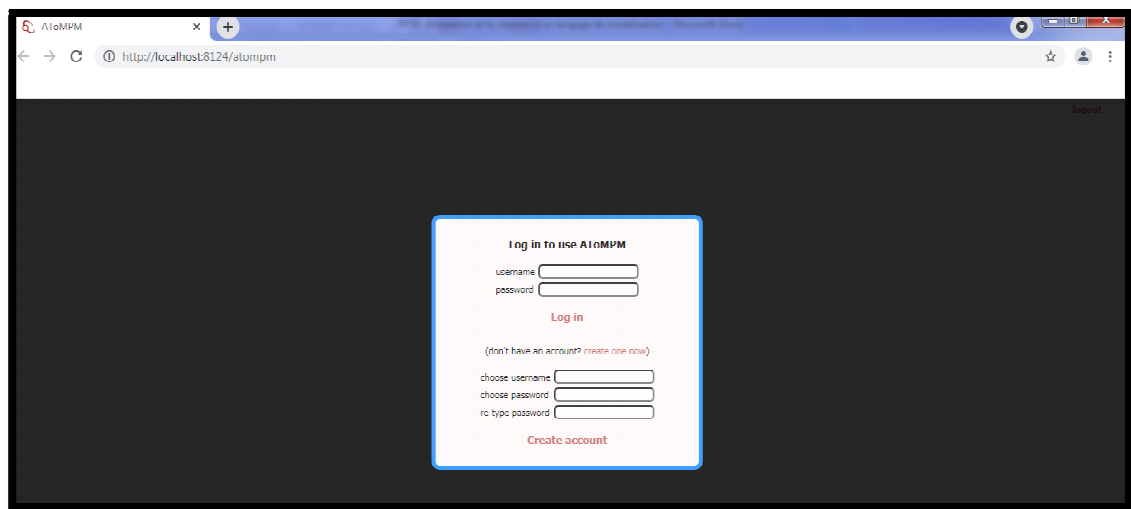


Figure1: ATOMPM create account.

Figure 2 represents the ATOMPM interface after creating an account or authenticating if you have an account.

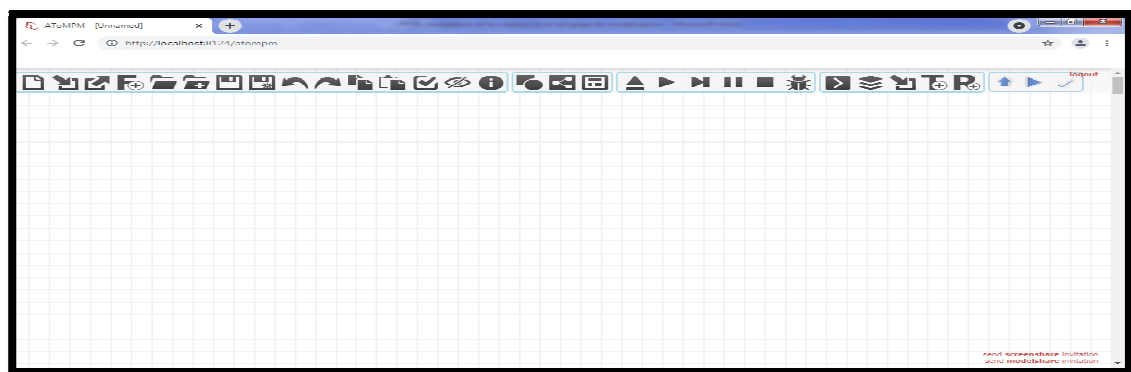


Figure2 : ATOMPM interface

TP 01: Creation of a modeling language for Petri Nets (PNs)

To initialize a new PN formalism, press the new formalism button (F+), from the main menu toolbar

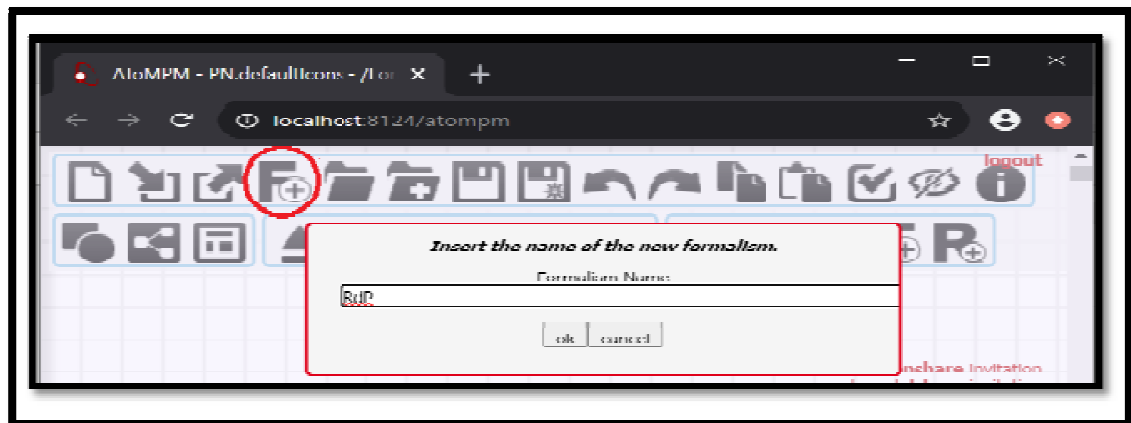


Figure03: new formalism.

This generates two files in the /formalisms/PN folder:

- ❖ PN.model: this is the abstract syntax of the PN model
- ❖ PN.defaultIcons.model: this is the concrete syntax of the PN model

1. Definition of the abstract syntax of PN(PN.model)

The first step is the construction of the meta-model for the Petri nets. To do this we use the UML class diagram (menu of the Simple Class Diagram formalism). Draw the PN's meta-model then save it in the PN.model file.

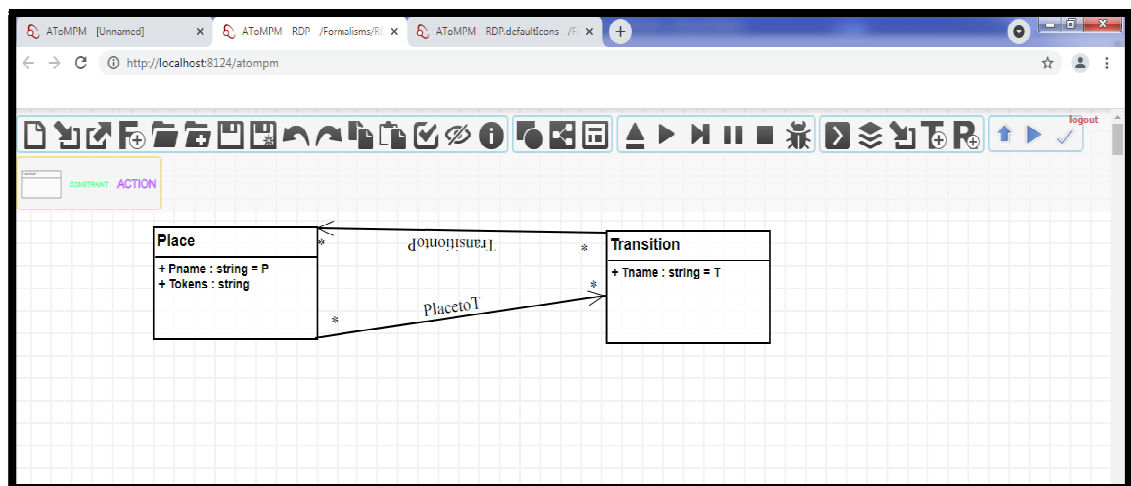


Figure 04: meta-model of PNs

Compile the PN.model for meta-model generation (the PN.metamodel file) using the

compile button.



TP 01: Creation of a modeling language for Petri Nets (PNs)

2. Definition of the concrete syntax of the PN (meta-model) PN

.defaultIcons.model:

Open the model (concrete syntax) of the PN in the /PN/PN.defaultIcons.model file.
Draw the concrete syntax of the PN then save it in the PN.defaultIcons.model file.

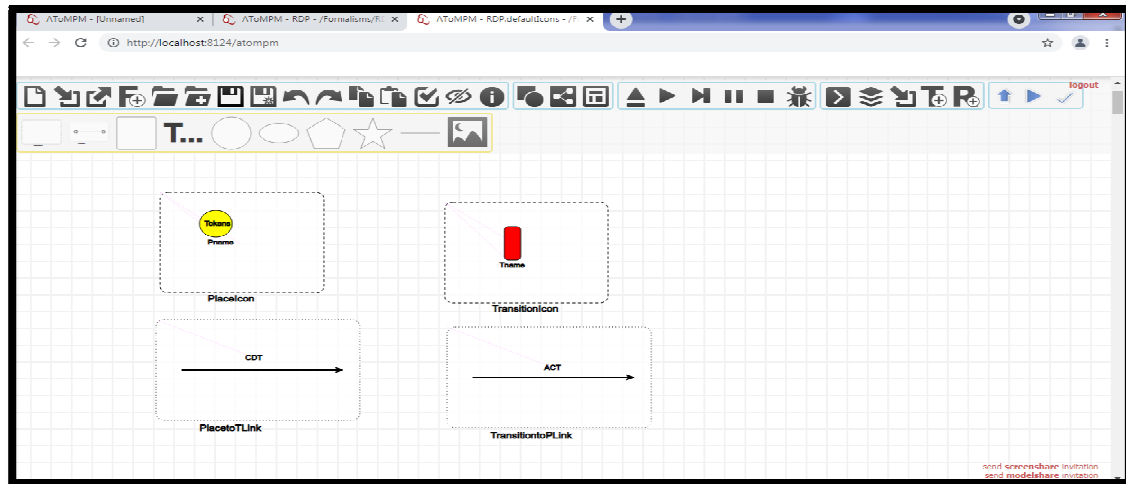


Figure 05: PN.defaultIcons.model

✓ The code mapper:

- Name place: ({textContent:getAttr("Pname")})
- Tokens: ({textContent:getAttr("Tokens")})
- Name of transition: ({textContent:getAttr("Tname")})
- Condition of transition: ({textContent:getAttr("CDT")})
- Action of transition: ({textContent:getAttr("ACT")})

Compile the RdP.defaultIcons.model for meta-model generation (the

RdP.defaultIcons.metamodel file) using the compile button.



3. Create and Use of a PN model

The first step when using a language is to load its associated formalism toolbar by pressing the second button of the Main Menu Toolbar. You are presented with a file browser. Load the file: **formalisms /RdP/RdP.defaultIcons.metamodel**, then create the following PN:

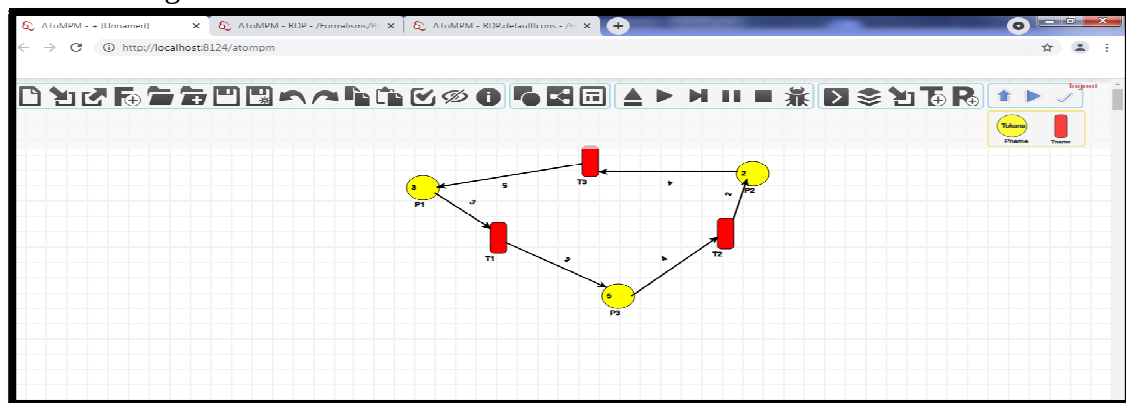


Figure06 : PN model