The ePNK: Hands-on / Project

Ekkart Kindler

DTU Compute
Department of Applied Mathematics and Computer Science
Task

MBSE f/w Petri Nets: Hands-on / project
Task

MBSE f/w Petri Nets: Hands-on / project
Task

Condition

MBSE f/w Petri Nets: Hands-on / project
Task

MBSE f/w Petri Nets: Hands-on / project

Declaration

```java
int x = 0;
```
Step 1: PNTD

- Implement a new Petri net type extending PTNets (a PNTD for what we call PNCode) by adding:
  - Action(label)s to transitions
  - Condition(label)s to transitions
  - Declaration(label)s to pages

- These concepts should be text in Java syntax
- For now, it is not necessary to check syntactical correctness of this syntax
- A basic project with a set up of a simple model is provided to you (see slides on details later)
Right-click (on type PNCode):

→ ePNK

→ Generate Java Code from PNCode
import org.pnml.tools.epnk.tutorials.pnmbse.codegenerator.runtime.AbstractPetrinet;

public class Second extends AbstractPetrinet {

    // PNCode declarations
    int x = 0;

    public Second() {
        super(new int[][][1], new String[]("t1", "t2", "t0", "t4");
    }

    public boolean enabled(Int t) {
        switch (t) {
            case 0:
                return marking[0] >= 1 && x < 5;
            case 1:
                return marking[1] == 1;
            case 2:
                return true;
            case 3:
                return marking[0] >= 1 && marking[1] >= 1;
            default:
                return false;
        }
    }

    public void fire(Int t) {
        if (!enabled(t))
            return;
        switch (t) {
            case 0:
                marking[0] = marking[0] - 1;
                System.out.println("x = " + x);
                x++;
                break;
            case 1:
                break;
        }
    }
}
Right-click:  
→ Run As  
→ Java Application

\[ x = 0 \]
\[ x = 1 \]
Step 2: Code Generator

- Implement an action that generates Java code from such PNCode running as a Java application

- The GUI and “runtime” environment as well as all to the set up the action and initiate the code generation is provided to you

- Also a generator template for the basic Petri net (not taking actions, conditions and declarations into account) is provided to you

- You can focus on extending the template and skeleton for generating the code for actions, conditions and declarations