École Centrale de Nantes

M2 - CORO-Embedded Real-Time Systems

# Modelling and Verification using Petri nets

*Authors:*
Ragesh Ramachandran
Gopi Krishnan Regulan

*Supervisors:*
Olivier (H.) Roux

January 24, 2019

# Contents

# Chapter 1

# Introduction

Petri nets are a tool for the study of the systems. The Petri nets theory allows a systems to be modelled by a Petri nets which is a mathematical representation of the system. Analysis of the Petri nets can reveal important information about the structure and dynamic behavior of the system. This information can be used for the evaluation of the system or and then make changes or improvements in the system.

The modelling in Petri nets is useful in cases when the system can not be studied directly and can only be studied through a mathematical representation. By representing a system as a mathematical model and converting the model into instructions for a computer, and running the computer it is possible to model larger and even complex systems then ever before.Petri nets are useful in the modelling of the systems with concurrency and synchronization.

In the practical session we will be using the Petri nets as an auxilary modelling and analysis tool. The modelling is done on Romeo which is a software studio for Time Petri Net analysis, developed in the Real-Time Systems Team at IRCCyN . It performs analysis on T-Time Petri nets and on one of their extension to scheduling. The modelling and verification of a product chain and a coffee machine is modelled in the romeo using Timed Petri nets.

# Chapter 2

# Modelling and verification of a production chain

## 2.1 Description of the system

1. **Product A :** Raw materials A ($rm_A$) are always available in the area **Raw materials A**. The **robot A** turns raw material A into product A ($p_A$) and drops them in the storage zone **ZA**. The manufacturing time of robot A is between 30 and 40 s. The storage capacity of product ($p_A$) in zone **ZA** is 3. When the zone **ZA** is full, the robot A stops until a place becomes available in the zone **ZA**.

2. **Product B :** Raw materials B ($rm_B$) arrive periodically every 60s. The **robot B** turns raw material B into product B ($p_B$) and drops them in the storage zone **ZB** in 30s. The storage capacity of product $p_B$ in zone **ZB** is only 1. If the robot B drops a product $p_B$ whereas the zone **ZB** is not empty, the obstruction causes an accident.

3. **Loading :** The loading robot loads, when the assembly cell is ready, a product $p_A$ and then a product $p_B$. It requires 5s to load a product.

4. **Assembling:** As soon as a product $p_A$ and a product $p_B$ are available in the cell, the cell assembles the products and drop the result on the conveyor belt for the evacuation. The assembly cell is then available to receive new products $p_A$ and $p_B$ . The assembling time is between 20 and 25s.

## 2.2 Modelling, verification, sizing

### 2.2.1 Modelling

The system is modelled using timed petri nets as per the given specifications. The

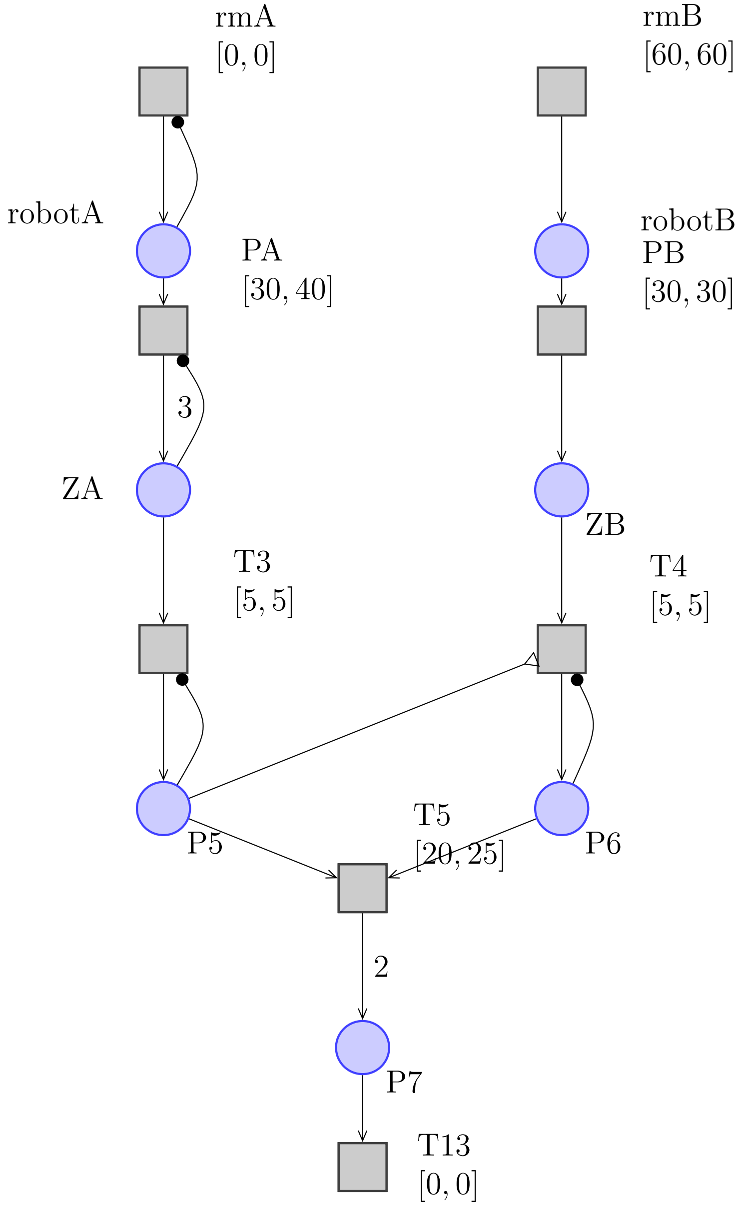



Figure 2.1: Petri nets model of production chain

```
AG markingBounded(3) and ZB <= 1 and robotB <= 1

Property is True
```

### 2.2.2 Verification

1. *We want to buy a cheaper (and therefore slower) assembly cell. How much can we slowdown the assembly cell without this being a problem (without obstruction in zone ZB) ?*

   The time limit for the transition present in the assembly cell is changed to$[25+a, 25+a]$. By simulating the system by checking the integer parameters in the control panel and the property given below is verified. The assembly can be slowed down to an interval of $[0, 25]$ without being a problem.

   ```
   AG ZB <=1

   answer : a in [0, 25]
   ```

2. *How much can we reduce the period of arrival of raw materials rmB in order to accelerate the production without causing obstruction in zone* ***ZB*** *?*

   The time limit of the trasnsition $rmB$ is changed from $[60, 60]$ to $[60, 60]$in order to evaluate the new time period of arrival of raw materials.By simulating the system by checking the integer parameters in the control panel the property given below is verified to determine the value of a. The vnew time period of arrival of raw materials to accelerate the production is determined to be **40 seconds**.

   ```
   AG (ZB <=1 and robotB <=1  and markingBounded(3))

   a in [40, inf[
   ```

# Chapter 3

# Modelling and verification of a coffee machine.

## 3.1 A first simple model

The machine can make coffee or tea. The cycle of the machine is

1. Waiting for a coin.
2. Waiting for the choice between coffee or tea.
3. Make the coffee or the tea corresponding to the choice.

### 3.1.1 Specification

Model the coffee machine with the following specification.

- When a coin is inserted, if nothing happens after 10 seconds, it is given back.
- A coin is given back if the user presses the refund button.
- When a piece is present in the machine, if a second piece is inserted, it is given back(refund) immediately.
- When a coin is present in the machine, the user may request a coffee or a tea.
- A coffee is made in 30 seconds. Meanwhile, no action is possible from the user.
- A tea is made in 25 seconds. Meanwhile, no action is possible from the user.
- An action will be possible again only when the cup is actually removed by the user.

### 3.1.2 Model

The system is modelled in Romeo and validated. The transitions in the system are :

- **insertCoin** when a coin is inserted.
- **Refund** when the customer is refunded.
- **ChooseCoffee** when preparation of the coffee starts.
- **ChooseTea** when preparation of the tea start.
- **CoffeeOut** when the coffee is ready.
- **TeaOut** when the tea is ready.
- **PressOK** to finish the preparation of coffee/tea.

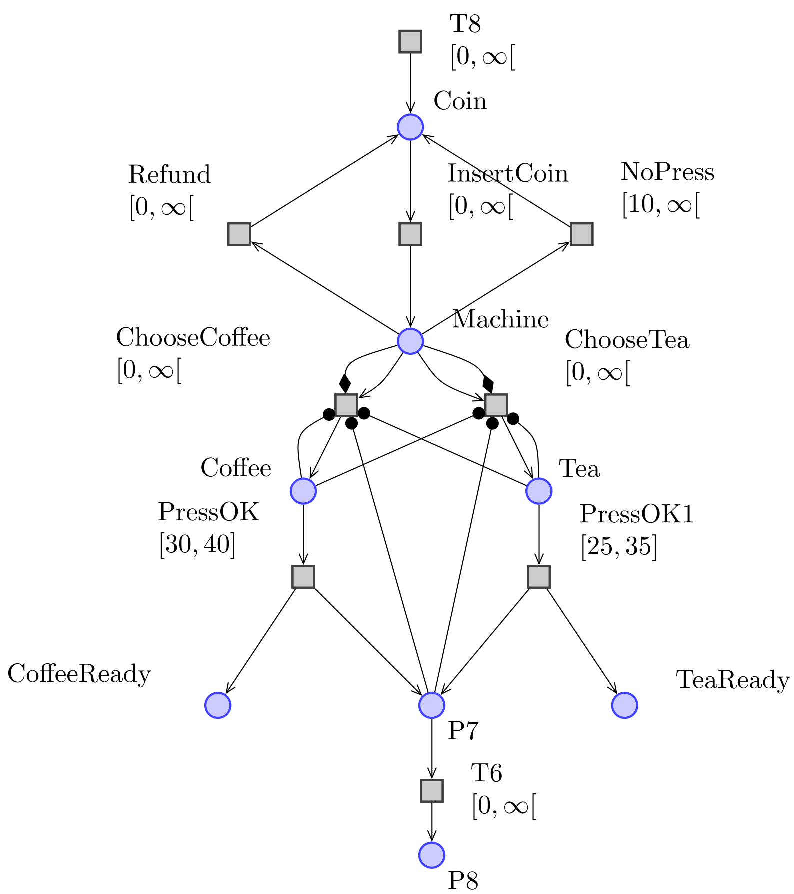



Figure 3.1: Petri nets model of coffee machine

The coins can be chosen by the user. But the machine accepts only one coin and all the other consecutive coin added will be refunded once the coffee or Tea is chosen. The user has to choose the coffee or Tea in 10 seconds. Otherwise, the coin is refunded. The user has also the option to choose the refund before choosing the coffee or tea. The coffee can be ready in 30 seconds and the tea can be ready by 25 seconds. After the user can take the coffee or tea out.

### 3.1.3 Verification

1. **Functional properties** such as if the customer asks for a coffee and does not ask for the refund then he gets a coffee is verified and is found to be true.

```
(Coffee == 1 and Coin == 0) -->  CoffeeReady == 1
```

2. **Verification of Boundedness**

   The system is not 1 bounded.

```
AG markingBounded(1)

false
Traces:
-> T8, InsertCoin, T8, Refund
-> T8, InsertCoin, T8, None
```

   The system is not 2 bounded.

```
AG markingBounded(2)

false
Trace: T8, InsertCoin, T8, ChooseTea, PressOK1,
InsertCoin, T8, ChooseTea, PressOK1,
InsertCoin, T8, ChooseTea, PressOK1
```

   The system is not 3 bounded.

```
AG markingBounded(3)

false
Trace: T8, InsertCoin, T8, ChooseTea, PressOK1,
InsertCoin, T8, ChooseTea, PressOK1, InsertCoin,
T8, ChooseTea, PressOK1, InsertCoin, T8,
ChooseTea, PressOK1
```

3. **Verification of the absence of deadlocks**

```
AG[0,inf](not deadlock)

True
```

## 3.2 Optimal cost

### 3.2.1 Model description

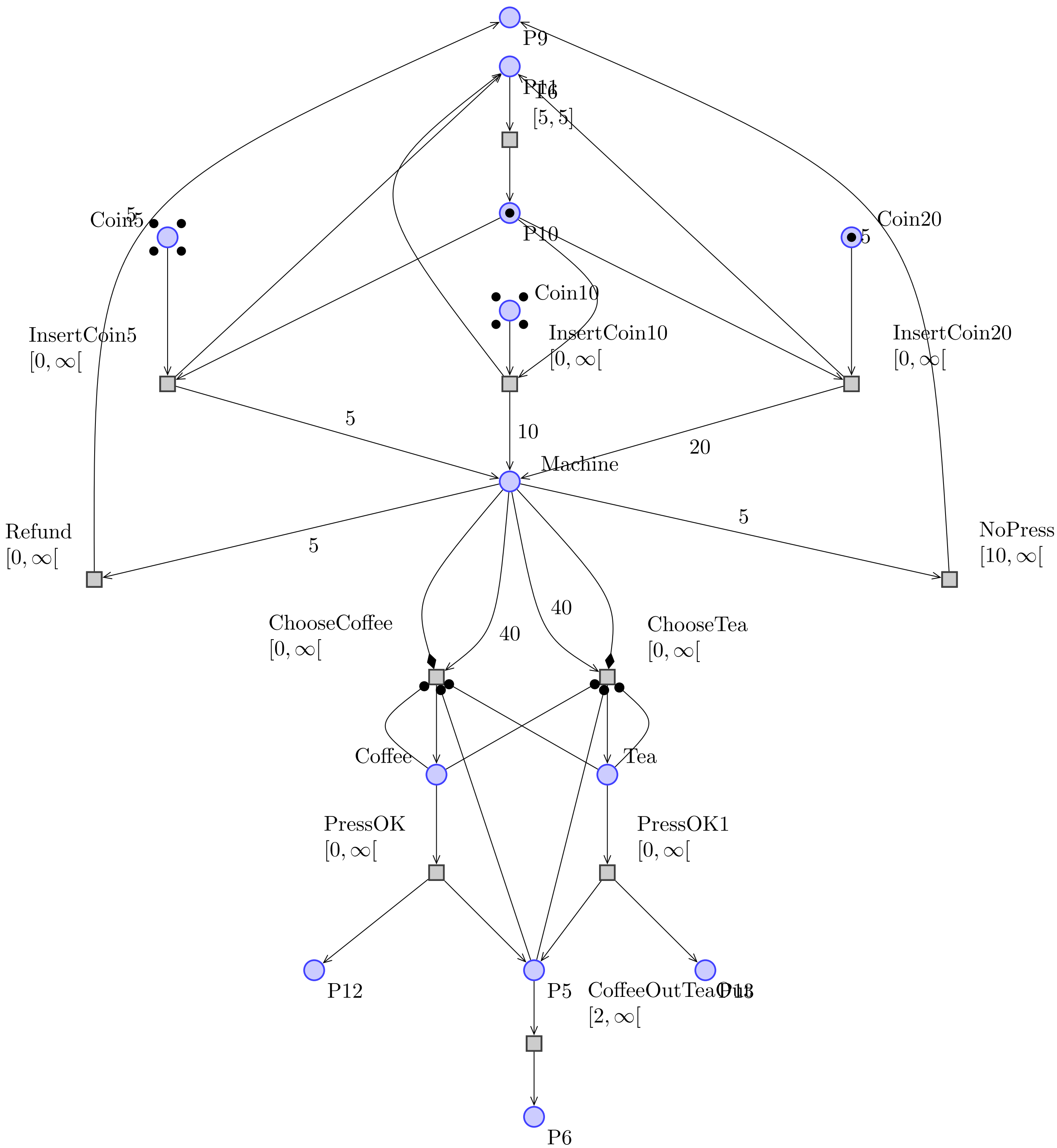
The model now takes into account the payment.

1. A drink (coffee or tea) costs 40 cts and the machine accepts 5, 10 and 20 cts but does not give change. The over payment is not memorised for the next cycle. The time between inserting 2 coins is at least 5s.
2. Once the user has chosen his drink and the preparation of the drink begins, a new cycle can start but the new choice of drink will be possible only when the cup is actually removed by the user.
3. It takes at least 2s for the user to take the cup.

A user wants a coffee and a tea and has 4 coins of 5 cts, 4 coins of 10 cts and 1 coin of 20 cts.Using the cost mode of Romeo, generate the optimal strategy for this user to get a coffee and a tea in minimal time.

### 3.2.2 Modelling with payment taken into account

The Petri nets model developed in the previous section has been modified to take payment into account. The cost of a drink is 40 cts and the denominations that are accepted in the machine are 5,10 and 20 cts. So this part is modelled by adding weights of *5,10* and *20* on the arcs which is connected to the place *Machine*. Between insertions of two coins a delay of 5 seconds is introduced in the model. Then the user can make selection of the drink using the transitions *ChooseCoffee* and within 30 seconds after the selection has been made the Coffee is prepared or if the transition *ChooseTea* is selected the Tea is prepared within 25 seconds. In the mean time the user cannot make any actions. Any new actions is made possible only after the user takes the drink from the machine.




The optimal strategy is generated in the Romeo for the user to get a coffee and tea in minimal time and the resulting cost is 42.

```
mincost(P12==1 and P13 == 1 and P6 == 2)
```

```
42

InsertCoin10, T6, InsertCoin10, T6, InsertCoin5, T6,
InsertCoin5, T6, InsertCoin10, ChooseTea, T6,
InsertCoin5, T6, InsertCoin5, PressOK1, TInsertCoin10,
T6, InsertCoin10, T6, InsertCoin5, T6, InsertCoin5,
T6, InsertCoin10, ChooseTea, T6, InsertCoin5, T6,
InsertCoin5, PressOK1, T6, InsertCoin20, T6, InsertCoin10,
T7, ChooseCoffee, PressOK, T7 6, InsertCoin20, T6,
InsertCoin10, T7, ChooseCoffee, PressOK, T7
```