Handout II - ICTiB/Process modelling

(Pieter Kwantes)

February 6, 2010

Contents

1 Exercises

1.1 Modelling with PT-systems

In this exercise you will create the Petri net model (a PT system) described in the book "Workflow Management. Models, Methods and Systems" chapter 2, exercise 2.5. a) with Woped.

1.1.1. First create a PT system that *abstracts* from resources, i.e. you exclude the instructors and examiners from the model. That means that the model will only represent the *route* that a student can follow through the process of the driving school. Make sure that the model meets the criteria for a *Workflow Net*.

1.1.2. Execute the model step by step using the *start tokengame* menu option of Woped.

1.1.3. Use the *Analyze* menu option of Woped to calculate the reachability graph (submenu *Reachability Graph*). Use this reachability graph to answer the question whether the Petri net model you created is sound or not. Explain your answer.

1.1.4. Use the submenu *Semantical analysis* in the *Analyze* menu of Woped to verify your answer of exercise 1.1.3. Try to explain all the results you find under the *Structural analysis* and *Soundness analysis* as given by Woped using the theory explained during the lecture.

1.1.5. Now add the resources to the model in accordance with the description in the exercise, i.e. 5 instructors and 3 examiners. Repeat the steps you did in exercises 1.1.2. through 1.1.4. Compare the results you find with the results that you found for exercises 1.1.2. through 1.1.4. and try to explain the differences.

1.2 Modelling with Workflow nets

Consider the business process of a car dealer, represented by the Petri net in figure 1. The process is initiated by the request of a customer who wants to buy a

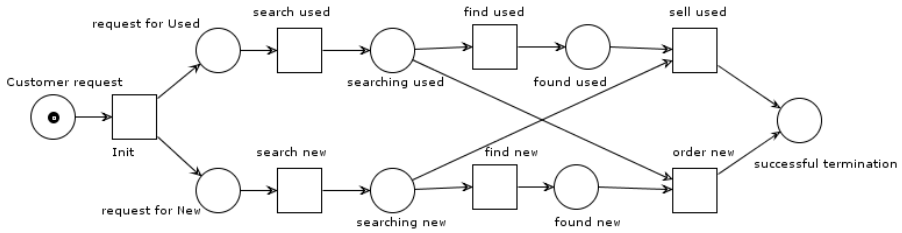

Figure 1: A PT-system representing the business process of a car dealer

car with some specification. The company has a department U for available used cars, and a department N for new cars that have to be ordered. We assume that, for each customer specification, a used car might or might not exist but a new car can always be found. Department U executes the tasks *search used* and *find used* and after finding a used car it executes the task *sell used*. Department N executes the tasks *search new* and *find new* and (after successful termination of that task) *order new*. To reduce waiting time for the customer, the tasks *search used* and *search new* are enabled concurrently after intake of the request by the service desk (task *init*). As soon as *find used* terminates successfully the task *sell used* is executed, which also involves disabling the task *find new*. Similarly, executing the task *order new* disables the task *find used*.

1.2.1. Explain the concepts *WF nets* and *soundness* of WF nets.

1.2.2. Explain why the Petri net in figure 1 is a WF-net.

1.2.3. Draw the reachability graph of the Petri net in figure 1.

1.2.4. Is the WF-net in figure 1 a sound WF-net? Why or why not? Give the violations of soundness, if any, and explain how these violations can be identified in the reachability graph you have drawn to solve Exercise 1.2.3.

1.2.5. If the WF-net shown in figure 1 is not a sound WF-net, change it in such a way that it *is* a sound WF-net, **without removing the parallelism between the search tasks of department U and department N**. Draw the changed (sound) WF-net and explain your solution.

1.2.6. Create the given model and the improved model in Woped. Then check your answers by analyzing the models with Woped using the submenus *Reachability Graph* and *Semantical analysis* like you did in exercise 1.1.3. and 1.1.4.. Explain the analysis results given by Woped.

1.3 Running case

During this course you will develop a Petri net model of the order process at the LISTA Company. You will not be using Woped for this but an other tool for modelling and analyzing Petri nets : CPN Tools. You will learn more about CPN Tools in the next exercise class. For the remainder of the exercise classes in this course you will be working with CPN Tools.

For now the assignment is to make a first sketch of a Petri net model of the order process at the LISTA Company based on the Dataflow diagram you made in exercise 1.5. in Handout I of the first exercise class. This first sketch of the Petri net model should be constructed as much as possible:

- In a *top-down* fashion. Just follow the top-down structure of the dataflow diagram you created.
- As a *PT system* (to be more specific: as a *sound WF-net*).

2 Place/Transition Nets

2.1 Introduction

Place/Transition-nets (PT-nets) are a generalization of EN-systems. In the EN-system each place can only be true or false. In PT-nets a place can have a marking of any natural number. The dynamics also changes. A transition can decrease the marking of its input places by any number and increase the marking of its output places also by any number.
A major advantage of this extension is that PT-nets allow a more compact representation of real world processes than EN-systems. For example, one place with two tokens in a PT-net may be used in stead of two places with each one token in an EN-system. If an EN-system has many repetitive features, including places with a similar meaning, this can be used to shrink the EN-system into a much more compact PT-net.
A major drawback of PT-systems might be that the state space can be infinite thus making the analysis of PT-systems much more difficult than the analysis of EN-systems. This section will provide a formal definition of a PT-systems.

2.2 Formal definition of PT-systems

In this section we give a formal description of the structure, the dynamics and the state space of a PT-system. The description is based on [2].
To give a formal definition of a PT-system, the mathematical concept of **multiset** is required. The difference between a set and a multiset is that an element in a set can occur only once, while an element in a multiset can occur more than once. A multiset m over a non-empty set V is defined as a function from V to the set of natural numbers N, i.e. $m : V \to N$. A multiset is represented by :

$$\sum_{v \in V} m(v)'v.$$

The set of all multisets over V is denoted by By V_{MS}. The set $\{m(v)|v \in V\}$ is the set of coefficients of the multi-set m, and $m(v)$ is the coefficient of v. Furthermore $v \in m$ iff $m(v) \neq 0$. A number of operations can be performed on multisets like, addition, subtraction, scalar multiplication, comparison and size. $(V_{MS}, +)$ is a commutative monoid (p. 69 in [2]). The sum over multi-sets is defined as follows:

Let a finite multi-set $m \in V_{MS}$, a commutative monoid $(R, +)$ and a function $F : V \to R$ be given. Then

$$\sum_{v \in m} F(v)$$

is used to denote the **sum** which has an addend for each element of m. When an element appears several times in the multi-set m, there is an addend for each of these appearances.

For a more detailed description see [2] section 2.1.

Structure of a PT-nets A PT-net is a tuple :

$PTN = (P, T, A, E, I)$ satisfying the following requirements :

1. P is a finite set of places.
2. T is a finite set of transitions such that $P \cap T = \oslash$.
3. $A \subset (P \times T) \cup (T \times P)$ is a finite set of arcs.
4. $E : A \to N_+$ is a weight function.
5. $I : P \to N$ is an initialization function.

A marking of a PT-net and is a multiset over P. Markings describe global states consisting of places with a multiplicity. In PTN the initialization function I defines the initial marking of the system.

Dynamics of a PT-nets The dynamics of the PT-net, PTN=(P,T,A,E,I), is described by the concepts defined below:

1. A **step** is a multiset over T.
2. A step Y is **enabled** in a marking M iff the following property[1] is satisfied:
$$\forall p \in P : \quad \sum_{t \in Y} E(p,t) \leq M(p)$$
3. When $t \in Y$ and Y is enabled, then t is enabled in marking M.
4. When $t_1, t_2 \in Y$, $t_1 \neq t_2$, and Y is enabled, then t_1 and t_2 are concurrently enabled in marking M.
5. When $Y(t) \geq 2$ then t is concurrently enabled with itself.
6. When a step Y is enabled in a marking M it may **occur**, changing the marking from M_1 to M_2, defined by :
$$\forall p \in P \quad M_2(p) = M_1(p) - \sum_{t \in Y} E(p,t) + \sum_{t \in Y} E(t,p)$$

[1] Note that the multiplicity, $Y(t)$, is omitted in the formula specifying this property. See also the definition of summation over multisets in the beginning of section 2.2

The first sum on the right side of the equation represents the tokens that are removed, and the second sum represents the tokens that are added by the occurrence of step Y. The marking M_2 is said to be **directly reachable** from marking M_1.

An example of a PT-net representing a producer-consumer process with two producers and two consumers is illustrated in figure 2 below: The PT-system in

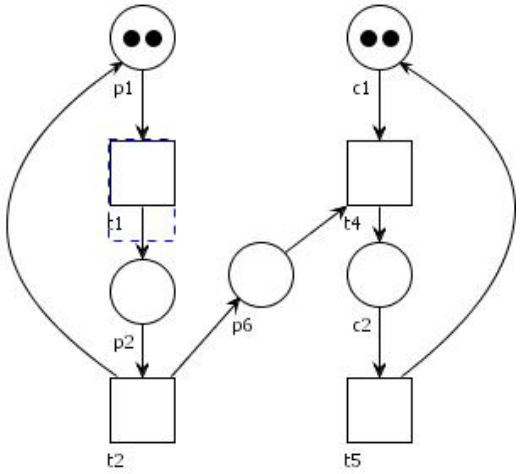

Figure 2: A PT-system of a producer consumer process with two producers and two consumers

figure 2 has the same places and transitions as the EN-system in figures 2 and 3 in the first handout but represents two consumers and two producers in stead of just one consumer and one producer. The number of consumers and producers in the PT-system can be increased simply by adding tokens to the places p_1 and c_1 in the initial marking. In an EN-system this would not be possible because each place can only contain one token.

Occurrence graph of PT-nets Analogous to the definition of an occurrence graph of an EN-system we can define an occurrence graph of a PT-system, the difference being that the graph might have an infinite size. (p. 115 [4]). There is however an algorithm to decide the finiteness of the occurrence graph.(p. 119 [4]).

References

[1] C. Girault, R. Valk; *Petri Nets for Systems Engineering*, Springer,2003

[2] K. Jensen; *Coloured Petri Nets; Basic Concepts, Analysis Methods and Practical Use; Vol. 1*, Springer-Verlag, 1992

[3] W. Reisig, G. Rozenberg, (Eds.); *Lectures on Petri nets: Basic Models*, *Lecture Notes in Computer Science*, 1491, Springer-Verlag, 1998

[4] G. Rozenberg, J. Engelfriet; *Elementary net systems*, in [3], pp. 12-121