Praktikum

Development of intelligent distributed embedded systems in mechatronics

Internship at a glance

Task "Day 1"

The task for the morning is mainly concerned with the question of how to model and analyze a control process. In the practice block on Petri nets, you will deal with the modeling and analysis of Festo systems in PIPE2. The aim is to model the control process of Festo systems in a Petri net and to make statements about the properties of this Petri net.

Task "Day 2"

In contrast to the other internship days, the tasks of day 2 are not divided into exercise and practice blocks. The task for the whole day is to make the sateflow of the machine in the simulation and connect it to the codesys and control the simulation stateflow by the code from codesys.

Task "Day 3"

On day 3 you will deal with object-related modeling and metamodeling. You will get an introduction to the corresponding tool (Obeo Designer) and will learn how to create metamodels and models with the help of this tool.

The aim is to create a finished service model for Festo systems at the end of the day, which is then used for model-to-text transformation the following day.

Task "Day 4-5"

After the tasks of day 3, you will learn how to perform model-to-text transformations using the Obeo Designer, i.e. how to transform data from the library model you have created (or from the model of its plant component) into a text file.

© Copyright AIS 2015

Day 1: Petri nets

1.1 Theoretical foundations

1.1.1 *Introduction Petri nets*

Petri nets are models of discrete, mostly distributed systems. They were first introduced by Carl Adam Petri in 1962, in his dissertation "Communication with automata". Petri nets are transition systems, i.e. they describe a system by means of states and their transitions. They are an excellent tool for modelling systems because they have simple and few elements and can be represented graphically well, which will be considered in the following.

1.1.2 *Formal definition of a simple Petri net*

In the following table, you will find the basic elements of a Petri net, as well as their formal definition.

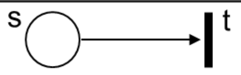
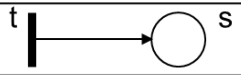
Description	**Formal Definition**	**Graphical representation**
Finite, non-empty **set of States S**	$S = \{s_1, s_2, \dots, s_n\}$	Circle or Ellipse ○ ⬭
Finite, non-empty **set of Transitions T**	$T = \{t_1, t_2, \dots, t_m\}$	Bars or Rectangles ▌ ▯
Finite, non-empty **set of Edges F (flux relation)**	$F \subseteq S \times T \cup T \times S$ $= Pre \cup Post$	Arrow →
Pre-Edge	$(s, t) \in Pre \subseteq S \times T$	
Post-Edge	$(t, s) \in Post \subseteq T \times S$	
Initial Marking **m₀** (Brand allocation) *a simple Petri network is understood here as a channel / instance network	$m_0 : S \rightarrow \mathbb{N}$ *Intuitively, a marker assigns the number of markers to the position*	Filled circle in one place ⊙

Fig. 1-1: Formal definition Petri net

The before and after areas of a transition are defined as follows:

- The preliminary areas of a transition t are all those places for which an **outgoing** precursor to this transition exists: $^{*}t = \{s \in S \mid (s, t) \in F\}$
- The after areas of a transition t are all those places for which an **incoming** mail edge from this transition exists: $t^{*} = \{s \in S \mid (t, s) \in F\}$

The switching rules of a condition/event net for a simple Petri net are

© *Copyright AIS 2015*

- A **transition** t is **activated** in a **marker** $\boldsymbol{m}$, if each **position** s contains a mark in the preliminary area:

$$m[t \Leftrightarrow \forall s \in \ {}^{*}t : m(s) = 1$$

- **Switches** a **transition** t. In this case, a mark is removed from the positions s in the preliminary area and a mark is added to each position s in the after area:

$$m'(s) = \begin{cases} m(s) - 1 & \text{falls } s \in \ {}^{*}t \\ m(s) + 1 & \text{falls } s \in t^{*} \\ m(s) & \textit{otherwise} \end{cases}$$

- If a **transition** t in one **mark** m is **switched** (status transition), this leads to a **subsequent marking** m**‘:** $m[t > m'$
  - $m[t > m'$means that the execution of the transition t in the marker m results in the marker m'.

1.1.3 *Verification of system properties*

Petri nets are used, among other things, for the verification of system properties, due to their good analyzability and simulability. The ultimate goal is the automatic verification of the properties by means of computer-aided model checking.

The following terms are used in model checking:

- **Termination**: There is an attainable mark in which the network cannot switch.
- **Liveliness**: Each transition can switch again and again (there is no point in time when a transition is never activated again).
- **No jamming**: There is no accessible mark where the mains cannot switch.
- **Reproducibility**: The initial mark can be reached from any mark.

1.1.3.1 *Termination*

Definition: There is an accessible marker where the network cannot switch:

$$\exists m \in M : \neg\exists t \in T : m[t \ \Leftrightarrow \exists m_1 \in M : \neg\exists m_2 \in M : (m_1, m_2) \in K$$

Interpretation:

1. An accessible mark m exists for which no transition t exists, which exists in m is activated.
2. An accessible mark m_1 exists that does not have a follow-on marker m_2 (subsequent marker: the tuple (m_1, m_2) is an element of the edge set K of the accessibility graph).

$\rightarrow$ **Every sequence of transitions reaches a deadlock at some point.**

© Copyright AIS 2015

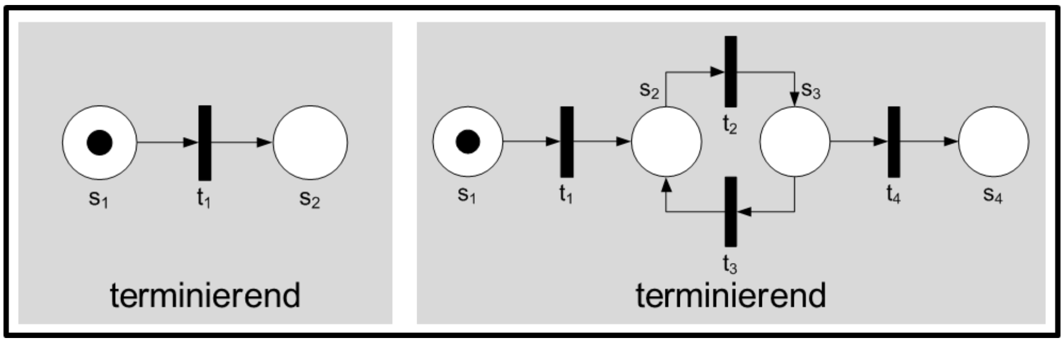


Fig. 1-2: Termination

1.1.3.2 *Liveliness*

Definition:

- Transition means
  - **Dead:** t is never activated ⟺ $\neg\exists m \in M : m[t$
  - **Activatable:** t is potentially activated (under at least one accessible marker) ⟺ $\exists m \in M : m[t$
  - **Alive:** t can be activated under any accessible marker
- Petri net means
  - **Dead**: all transitions are dead, i.e. $\forall t \in T : \neg\exists m \in M : m[t$
  - **(Weak) alive** (not jammed): under each marker, at least one transition is activated, i.e. $\forall m \in M : \exists t \in T : m[t$
  - **(Stark) alive**: all transitions are alive, i.e. $\forall t \in T : \exists m \in M : m[t$

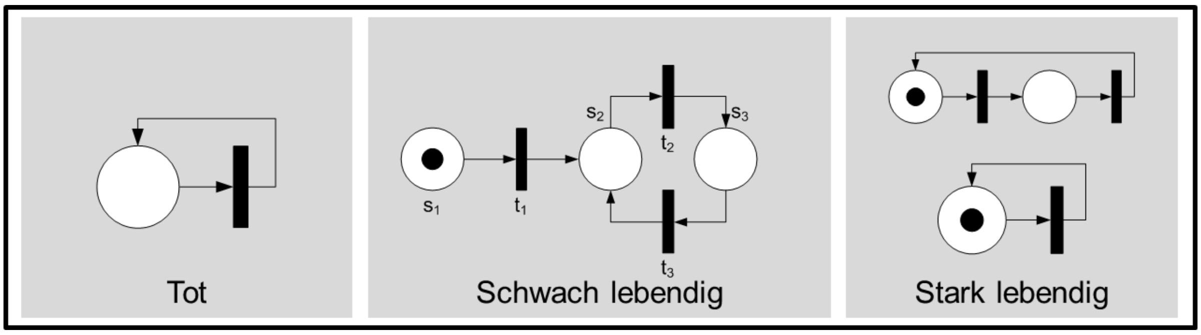


Fig. 1-3: Liveliness

© *Copyright AIS 2015*

1.1.3.3 *No jamming*

Definition: There is no reachable mark where the network cannot switch:

$$\forall m \in M : \exists t \in T : m[t \Leftrightarrow \forall m_1 \in M : \exists m_2 \in M : (m_1, m_2) \in K$$

Interpretation:

1. For each reachable mark m a transition t exists which in m is activated
2. For each reachable mark m_1 there is a subsequent marker m_2 (Follow-up mark: The tuple $(\mathrm{m}_1, \mathrm{m}_2)$ is an element of the edge set K of the accessibility graph).

→ **Termination and freedom from jamming are mutually exclusive.**

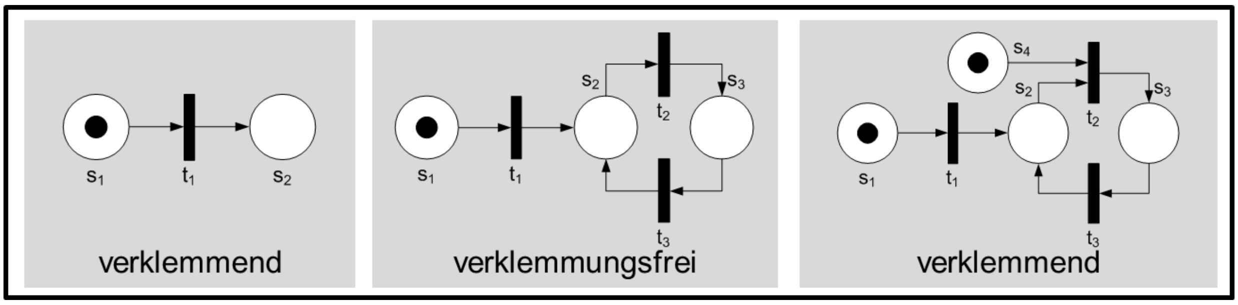


Fig. 1-4: Freedom from jamming

1.1.3.4 *Reproducibility*

Definition: The initial marker is accessible from any marker:

$$\forall m \in M : \exists n \in \mathbb{N} : (\forall i \in \{x \in \mathbb{N} | x < n\} : (m_i, m_{i+1}) \in K) \wedge (m_n, m_0) \in K$$

Interpretation:

1. For each reachable mark m there is a switching sequence that results in the initial marker.
2. This switching sequence consists of n transitions, where the tuple (m_i, m_{i+1}) is an element of the edge set K of the reachability graph for $i = 1, \ldots, (n-1) \Leftrightarrow i \in \{x \in N | x < n\}$ and the tuple (m_n, m_0) is an element of the edge set K of the reachability graph.

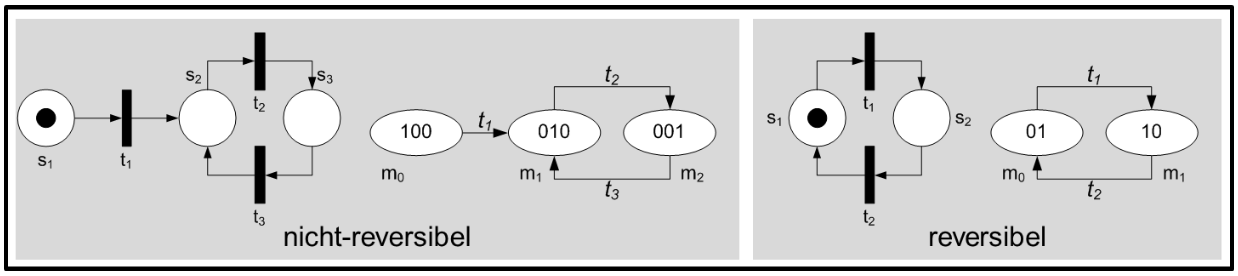


Fig. 1-5: Reversibility

© Copyright AIS 2015

1.1.3.5 *Reachability graph*

The reachability graph is a directed graph that can be obtained from a Petri net and an initial marker and is used to analyze the Petri net. A reachability graph can only be fully calculated for limited nets.

The set of M nodes of the reachability graph are all markings that can be reached by a Petri net. The edges of the reachability graph are tuples $(m_1, m_2) \in M \times M$ if there is a transition t which is in m_1 can be switched and m_2 is the subsequent marker, if the transition t in m_1 switches:

$$K = \{(m_1, m_2) \in M \times M \mid \exists t \in T : m_1[t \wedge m_1[t > m_2\}$$

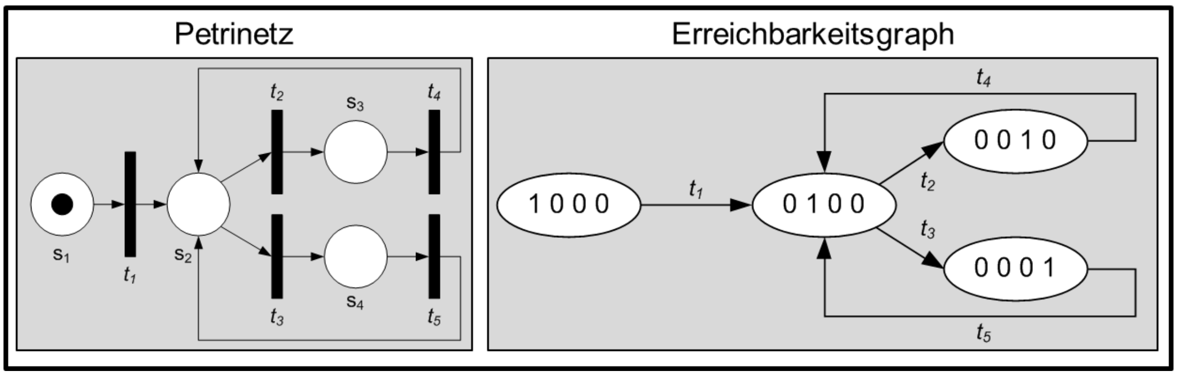


Fig. 1-6: Example accessibility graph

With the help of the reachability graph, it is easy to see, among other things, whether a Petri net is reversible or not.

© Copyright AIS 2015

1.1.4 *Vectorial calculation*

Petri nets can be analyzed not only by graphical methods but also by a vectorial representation. The basic elements of the vectorial representation of the switching behavior are the **transition vector**, the **incidence matrix,** and the **parikh vector**. With the help of these, a vectorial calculation of the subsequent marking can be carried out.

1.1.4.1 *Transition vector*

The switching behavior of a transition t can be described by means of a transition vector (markers are subtracted from places in the pre-zone; markers are added to places in the post-zone)
The dimension of the transition vector corresponds to the number of places $|S|$ in the petri net:

$$\vec{t_1} = \begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \end{pmatrix} \qquad \vec{t_2} = \begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \end{pmatrix} \qquad \vec{t_3} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ -1 \end{pmatrix}$$

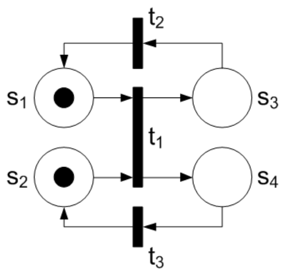


Figure 1-7: Example Transition Vector

1.1.4.2 *Incidence matrix*

The incidence matrix C reflects the switching behavior of the entire network:

$$C = \begin{array}{c} \begin{array}{ccc} t_1 & t_2 & t_3 \end{array} \\ \begin{pmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{pmatrix} \end{array} \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \end{array}$$

Fig. 1-8: Further example of incidence matrix from Petri net Fig. 1-7

1.1.4.3 *Parikh vector*

The parikh vector describes how often a transition is executed in a switching sequence.

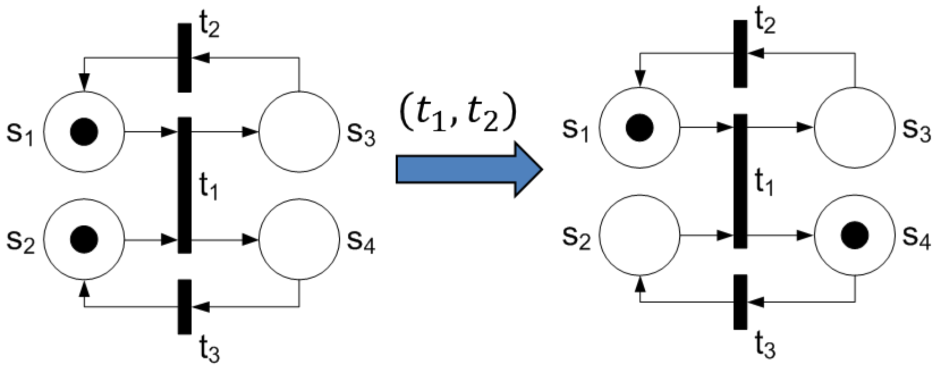


Fig. 1-9: Example parikh vector/switching sequence

The parikh vector for the switching sequence shown above (Fig. 1-9) would read accordingly:

$\vec{v} = (1 \quad 1 \quad 0)$

1.1.4.4 *Calculation of brand occupancy*

For each marker m there is a marker vector $\vec{m} = (m(s_0) \quad m(s_1) \quad \cdots \quad m(s_{|S|}))^T$
In a given marker m (Fig. 1-8), a given switching sequence is executed (represented by parikh vector).

$$\vec{m}' = \overrightarrow{m_0} + C \cdot \vec{v}^T$$

$$\vec{m}' = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{pmatrix} \cdot (1 \quad 1 \quad 0)^T$$

$$= \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ -1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Fig. 1-10 Example switching sequence

© Copyright AIS 2015

1.2 Introduction of the PIPE2 tool

Important: Documents, tools, tasks and sample solutions are made available on Moodle.

Symbol/ Icon	Meaning
○	Inserting States
▮	Inserting Transitions
↖	Inserting Connections
●✚	Adding Marks to the states
●✖	Remove the Marks from the States
🎨	Definition of different Colors (Colored Petri nets)
🚩	Switching between Animation and Editing Modes

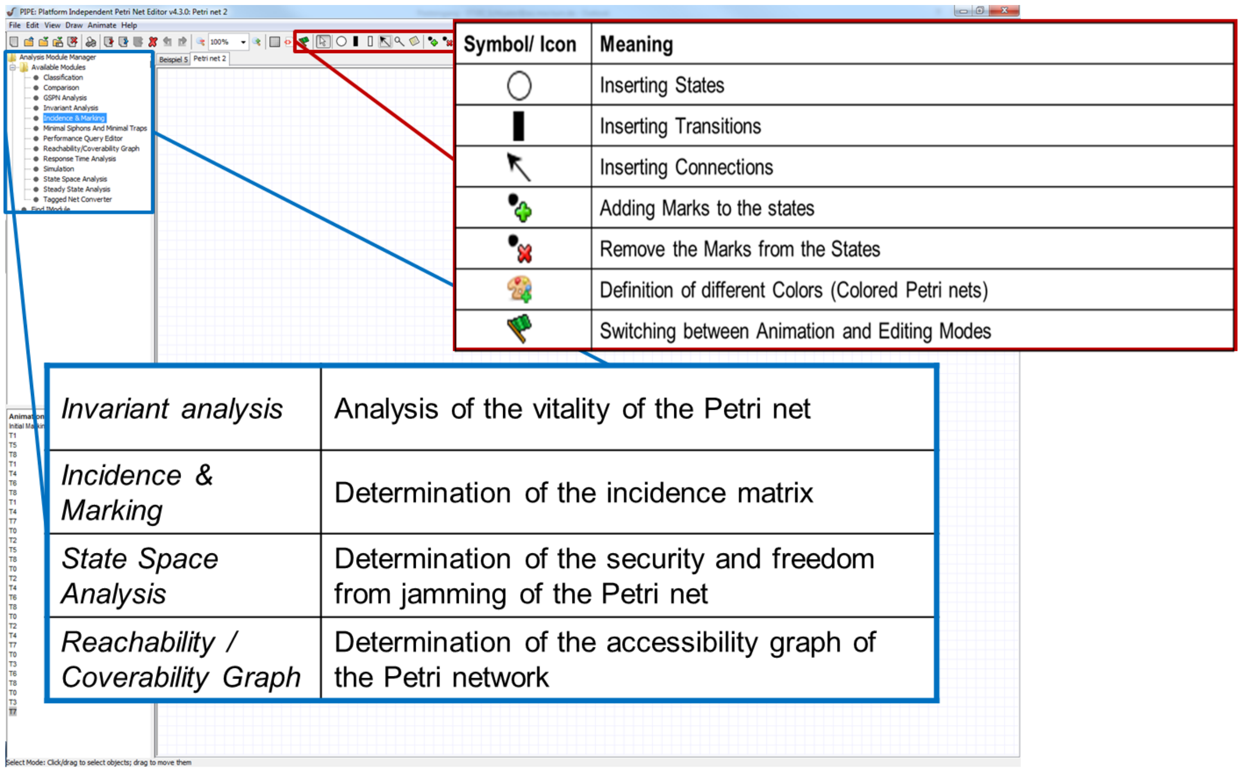
Fig. 1-11: Main icons of PIPE2

Basic functionalities of PIPE2

- **To start PIPE2**: Click the .BAT file *launch.bat* on the virtual machine desktop. (Note: It takes a few moments for PIPE2 to open)
- **Adding of states, transitions and marks**: Select icon → Add position, transition or marker in the graphical editor
- **Color definition**: Activate color and name it → Select color in the pull-down menu → Create connections or marks
- **Animation mode**: Icon Select animation mode → click to switch switch switchable transitions (red)
- Analysis **functionalities** (Analysis Module Manager)
  - *Invariant Analysis*: Analysis of the liveliness of the Petri net
  - *Incidence & Marking:* Determination of the incidence matrix of the Petri net
  - *State Space Analysis*: Determination of the safety and freedom from jamming of the Petri net
  - *Reachability/ Coverability Graph*: Determination of the reachability graph of the Petri net

Important: PIPE2 is a research tool. When working on the tasks remember to save your progress in the EiveSiM directory!

© Copyright AIS 2015



Symbol/ Icon	Meaning
○	Inserting States
▮	Inserting Transitions
↖	Inserting Connections
●+	Adding Marks to the states
●✗	Remove the Marks from the States
🎨	Definition of different Colors (Colored Petri nets)
🚩	Switching between Animation and Editing Modes

Invariant analysis	Analysis of the vitality of the Petri net
Incidence & Marking	Determination of the incidence matrix
State Space Analysis	Determination of the security and freedom from jamming of the Petri net
Reachability / Coverability Graph	Determination of the accessibility graph of the Petri network

Fig. 1-12: Overview of the user interface

© Copyright AIS 2015

1.3 Tasks

1.3.1 *Exercises*

Petri nets are an excellent tool for the graphical representation of processes by means of states and transitions.
During the internship you will carry out plant modelling. With the help of Petri nets you will visualize the functionalities, processes and states of the plant.
By analysing and simulating the system in advance, possible critical points, deadlocks or undesirable system states can be identified and avoided in advance.
In the following, the basic elements of Petri nets for system analysis are introduced, so that you can then carry out a plant modelling. The task for the morning is mainly concerned with the question of how to model and analyse a control process.

© Copyright AIS 2015

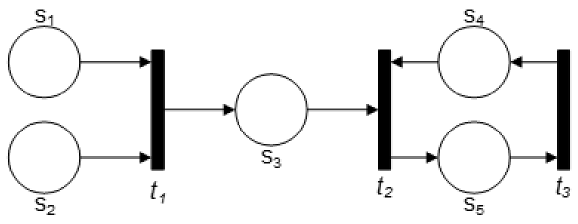
1.3.1.1 *Task 1*

Recapitulate the introduction to the subject of Petri nets. Then answer the following questions:

a) What are the basic elements of a petri net? What do Petri nets describe?

b) Network properties: When is a Petri net jam-free, alive, reproducible, terminating?

1.3.1.2 *Task 2*



Create and complete the following TOUs with the following initial markers for the given B/E network. Note that the capacity of each state is only one mark. The following table is an example for a), please generate the same table for b) and c):

a) s1, s2, s3, s5

b) s1, s2, s4

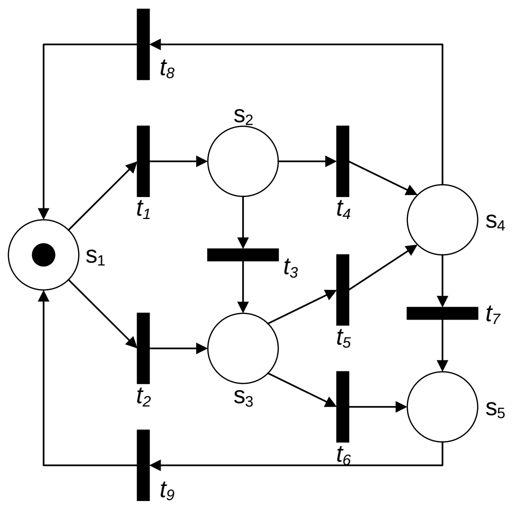
c) s1, s2, s5

marks in places   Transition switchable?

s1	s2	s3	s4	s5	t1	t2	t3
X	X	X		X			X
X	X	X	X			X	
…	…	…	…	…	…	…	…

Check your result using the PIPE2 software tool.
To do this, transfer the above Petri net into PIPE2 and use the simulation functionality of PIPE2.

1.3.1.3 *Task 3*



© Copyright AIS 2015

Given is the above Petri net with the initial mark $\overrightarrow{m_0} = (1 \quad 0 \quad 0 \quad 0 \quad 0)^T$.

Note: Please note that for the given B/E network in PIPE2 you have to set all job capacities to 1.

a) First determine the reachability graph manually and then verify your result with PIPE2.

b) First determine the incidence matrix manually and verify the result in PIPE2.

c) Calculate from m_0 the marker for the given switching sequences (Parick vectors). Use PIPE2 in the first step and verify your result by a corresponding calculation in MATLAB.
First switching sequence: $t_1 \rightarrow t_3 \rightarrow t_5 \rightarrow t_7 \rightarrow t_9 \rightarrow t_2 \rightarrow t_6 \rightarrow t_9 \rightarrow t_1 \rightarrow t_4$.
Second switching sequence: $t_1 \rightarrow t_3 \rightarrow t_5 \rightarrow t_6 \rightarrow t_7 \rightarrow t_9 \rightarrow t_9 \rightarrow t_2$.
What do you notice?

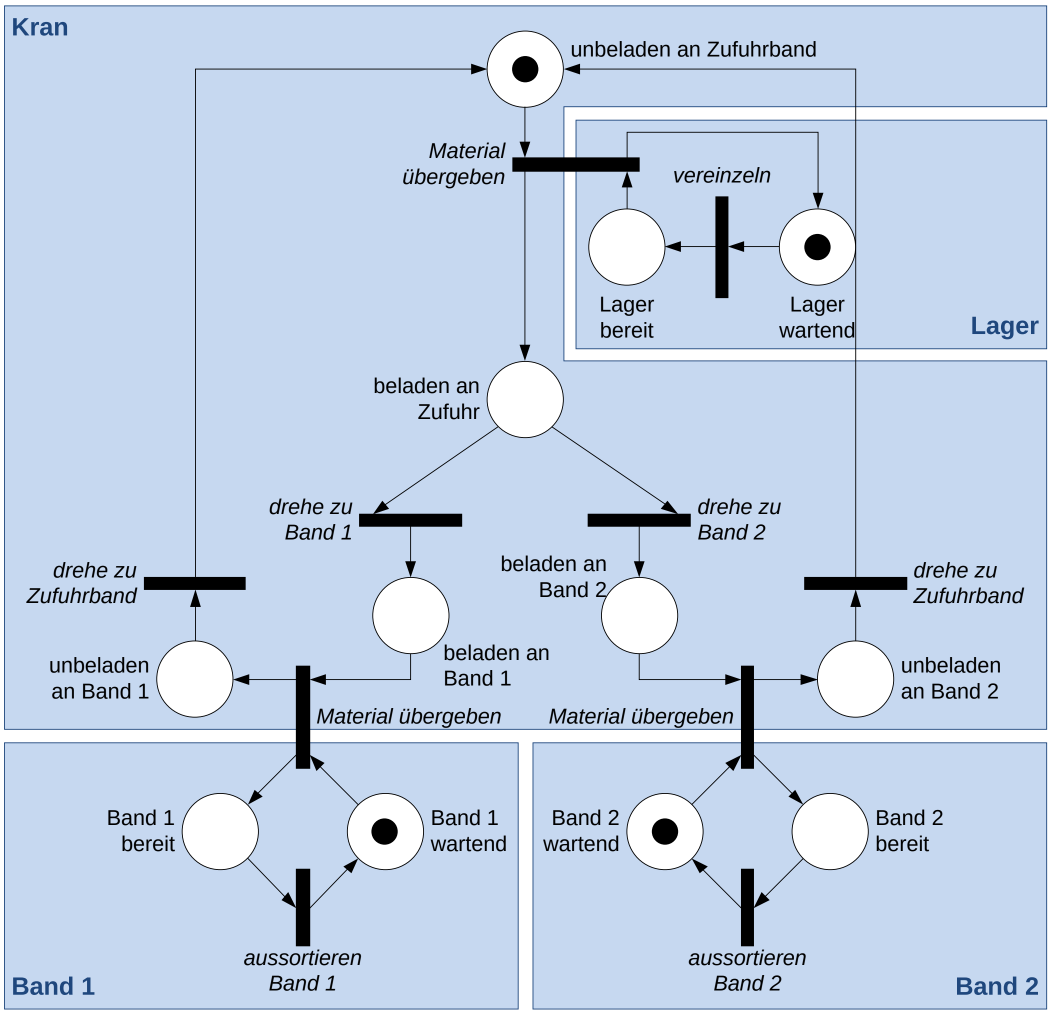
d) Analyze the reproducibility of the network.

© Copyright AIS 2015

1.3.1.4 *Task 4*

Two different types of workpieces are fed via a feed belt. A handling robot is to set down the workpieces alternately on belt 1 or 2. The following petri net is given.

Find the error! How must the petri net be modified according to the specification?



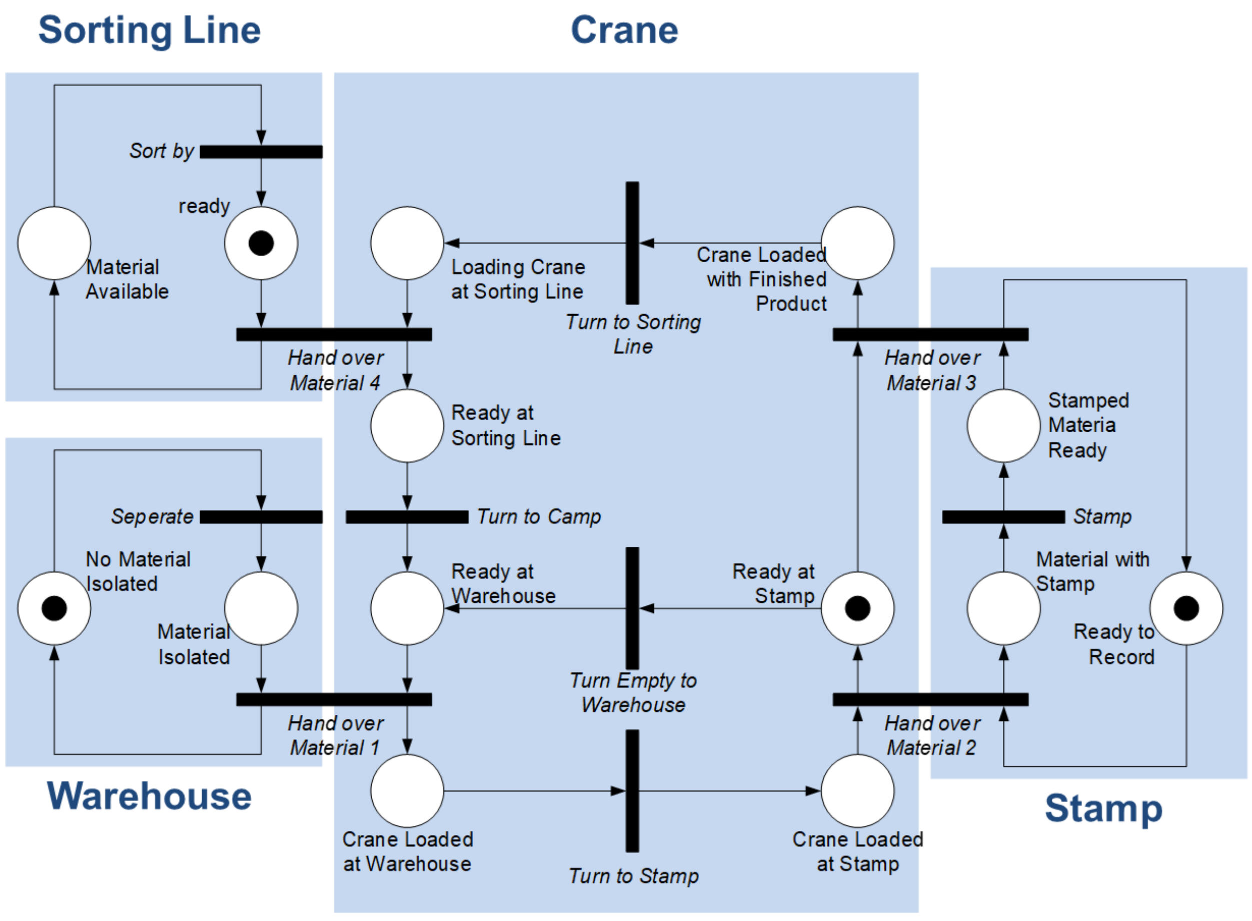
© Copyright AIS 2015

1.3.1.5 *Task 5*

Given is the Petri net of the stamping machine from the introduction to the topic Petri nets. The stamping plant consists of a warehouse with a separating stop from which workpieces can be removed, a stamp and a crane which transports the workpieces from the warehouse to the stamp or to the sorting line. Due to the model abstraction the petri net is not free of jamming. You will find the information in the folder EiveSiM Documents > Day 1 - Petri nets > Information > Task5.xml.

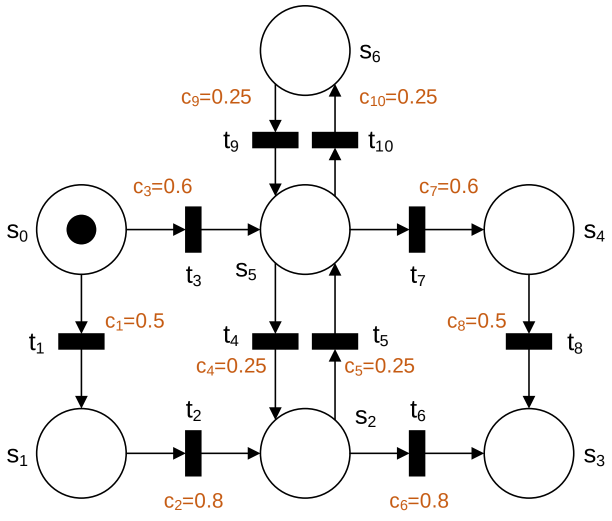
a) The petri net is jamming. Use the analysis functionalities of PIPE2 to verify this statement. What is the physical significance of this for the stamping plant?

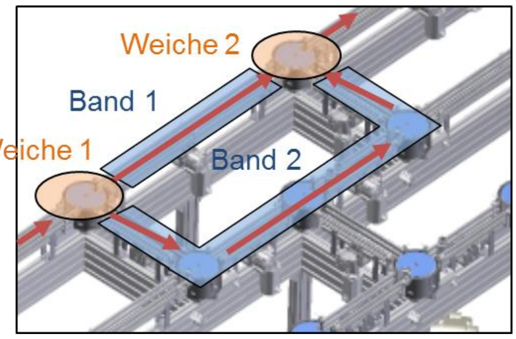
b) What are the possible solutions to overcome this deadlock? Verify the correctness of a solution with PIPE2. What is the physical significance of this for the stamping line?



© Copyright AIS 2015

1.3.1.6 *Task 6*





Given is the shown time-based, non-deterministic Petri net for a band system. It describes the network of a bottling plant consisting of conveyor belts and switches.

The positions correspond to possible bottle positions. The bottles can be transported further via different transitions (= conveyors). This transport takes different times, represented by the cost factors.

In this task, the optimum transport path of a bottle from the initial mark s_0 up to the end mark s_3 can be determined.
The specification and the OptimalSequence can be found in the folder EiveSiM Documents > Day 1 - Petri Nets > Specifications > Task6.xml or OptimalSequence.

Proceed as follows:

a) Determine the incidence matrix in PIPE2

b) Determine the cost vector (column vector of the cost factors).

c) Specify the start and end markers.

d) Formulate the optimization problem. Solve the optimization problem in MATLAB To do this, complete the OptimalSequence( ) function.
Note: Use the linprog command in MATLAB (see also: http: Note that to solve the correct problem, the solutions must be restricted so that only integer positive solutions are allowed ($\vec{x} \in \mathrm{N}^{|T|}$, $\vec{x} \geq 0$).

e) Verify your result.

© Copyright AIS 2015

1.3.2 *Acceptance of exercises Day 1*

Please hand in all exercises from day 1 to the last day of the internship in written form.

- Task 1
- Task 2: TOUs a),b) & c)
- Task 3:
    a) Availability graph in written form
    b) Incidence matrix matched with Pipe2 in written form
    c) Calculated end mark for switching sequence 1&2→ Useful?
    d) Reproducibility: Yes/No
- Task 4: Error description + at least 1 modification proposal (with drawing of the modified petri net)
- Task 5: Error description + at least 1 modification proposal (with drawing of the modified petri net)
- Task 6:
    a) Incidence matrix
    b) Cost vector
    c) Start and end marking
    d) Result of the optimization problem

© Copyright AIS 2015

1.3.3 *Practice block*

In the practical block on Petri nets, you will deal with the modelling and analysis of Festo systems in PIPE2. The aim is to model the control process of Festo systems in a Petri net and to make statements about the properties of this Petri net. Proceed as follows:

1. Familiarise yourself with Festo systems. The description of the Festo systems (see exercise script), which contains the sensor and actuator lists, the control process to be implemented and the respective requirements (under Acceptance), will help you do this. The exercise script can be found in the folder EiveSiM Documents > Exercise Script.
2. Identify the essential operations and also transfers (synchronisation and parallelization processes) of the Festo systems and enter them in your sketch.
3. Develop the Petri net for the control process in PIPE2 (Note: Consider which type of Petri net is suitable for the Festo systems and decide on a suitable level of abstraction for the representation)
4. Verify your petri net using the analysis and simulation functionalities provided by PIPE2
5. Analyse the characteristics of your petri net (liveliness, reversibility, freedom from jamming, termination). How can you influence these properties and what is the significance of these changes?
6. Document your results and prepare a mini-presentation on your findings.
7. Is your petri net deterministic or non-deterministic? How can you determine the best possible control process if your petri net is non-deterministic? Develop a procedure.
8. (*additional task*) Optimize your petri net in terms of comprehensibility, for example by formulating a colored petri net (predicate / transition net).

1.3.4 *Acceptance of practice block day 1*

- Complete Petri net of the respective plant component
- mini-presentation on your findings

© Copyright AIS 2015