Petri nets in industrial informatics

Laboratory of Modelling, Simulation and Control

Laboratory of Autonomous Mobile Systems

Petri nets, systems, and informatics

Gašper Mušič

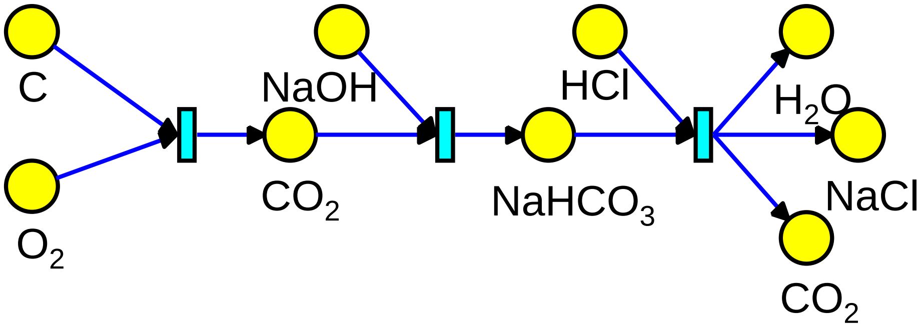
University of Ljubljana
Faculty of Electrical Engineering

Contents

- History of Petri nets
- What are Petri nets?
- Petri net theory
- Link to industrial informatics
  - control logic design
  - operations scheduling
- Conclusions

History of Petri nets

- Carl Adam Petri (1926–2010)
  - mathematician and computer scientist
  - born in Leipzig
  - invented the initial form of nets at the age of 13
    - for the purpose of describing chemical processes



  - as a teenager acquainted with the works of Einstein, Heisenberg, and German computing pioneer Konrad Zuse

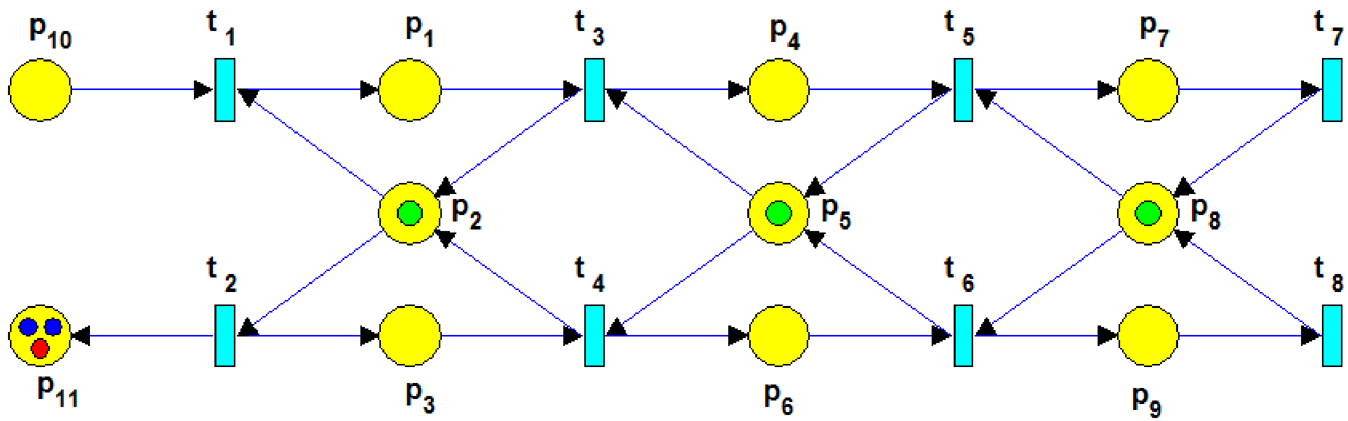
History of Petri nets

LMSC
LAMS

- end of 50s – work in Hannover and Bonn
  - work on theoretical framework for basic computing problems
    - Turing‘s model of infinite tape is physically infeasible
    - need for distribution of states and locality of interactions
- 1962 – dissertation Kommunikation mit Automaten at the Technical University Darmstadt
  - algebraic aspects of discrete concurrent (distributed) systems
  - sketch of a general net theory, proposal of distributed systems theory that complies to basic laws of physics
- 1968-91 – leader of information systems research institute within GMD (Gesellschaft für Mathematik und Datenverarbeitung)
  - development of Petri nets theory

History of Petri nets

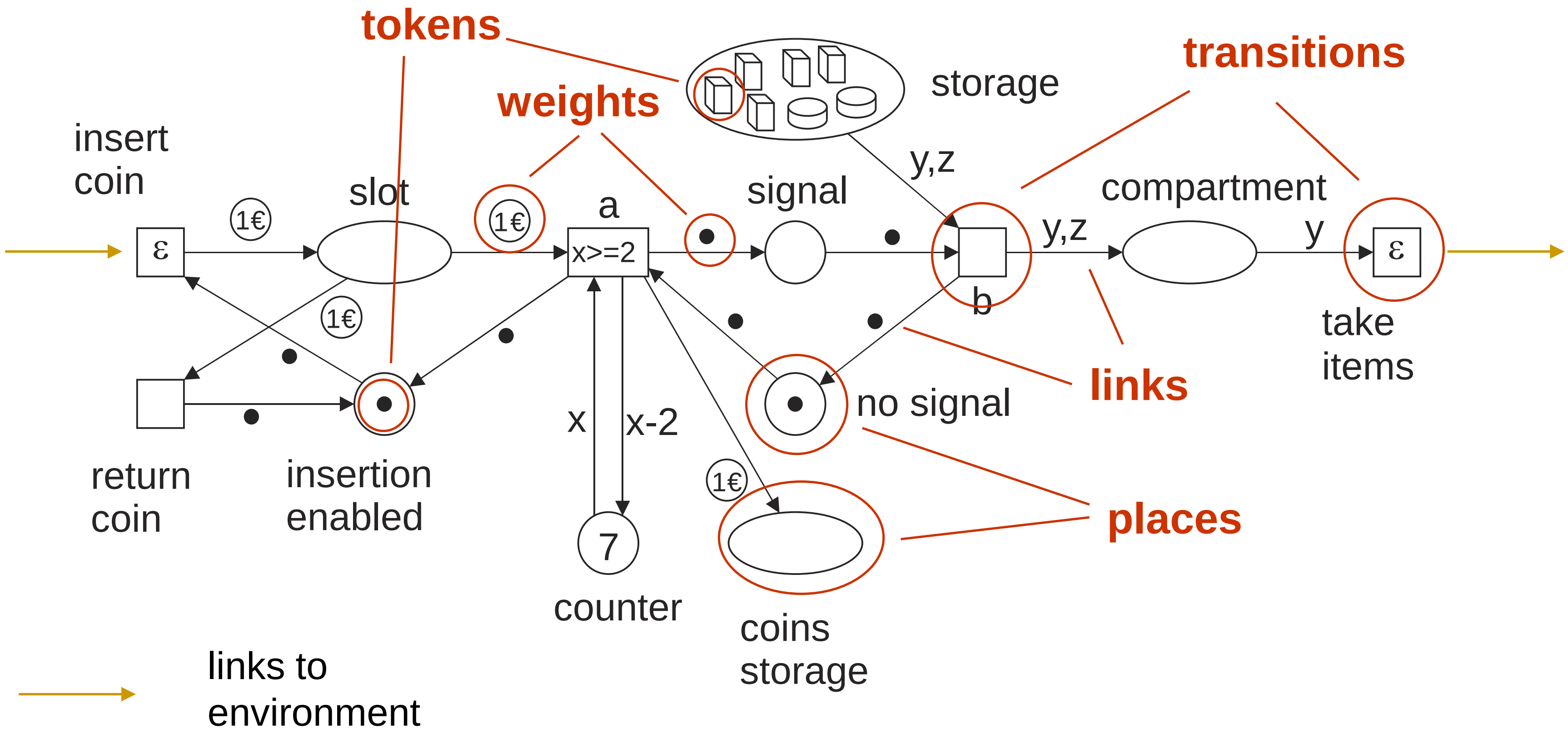
- Petri‘s model of expandable stack



modularity, expandability, asynchronity, locality of interactions

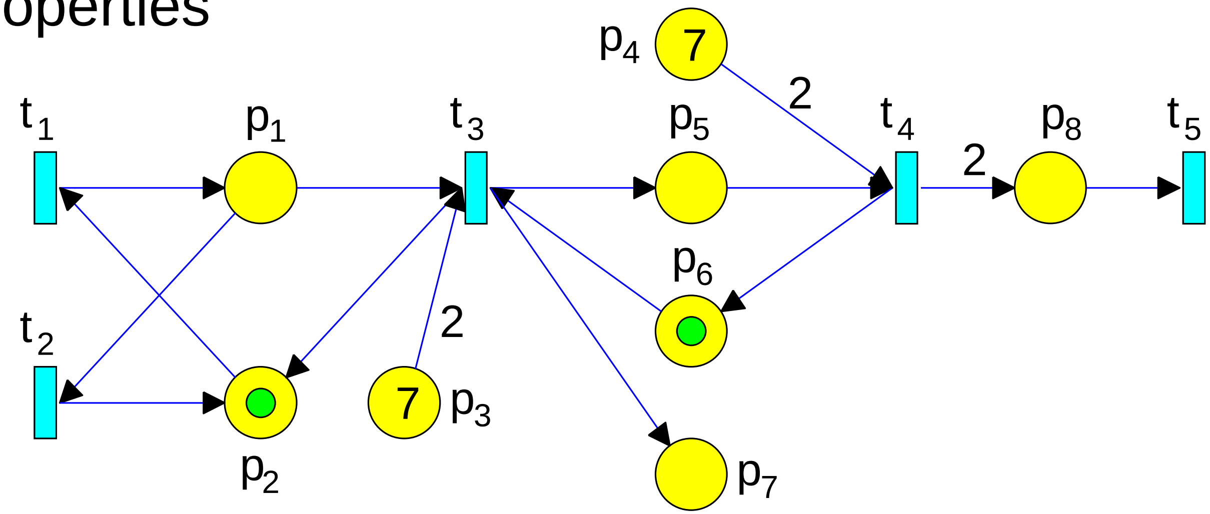
What are Petri nets?

- Model of activities and activity triggering conditions
- Example: vending machine



Place transition nets

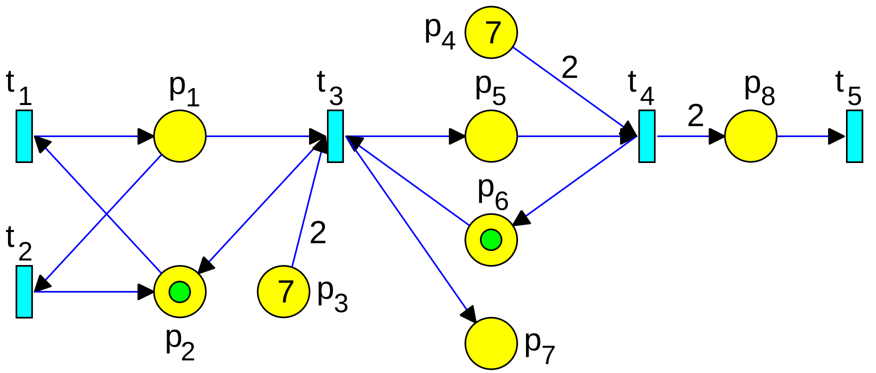
- Petri‘s original nets
  - only transitions, conditions, and arcs
- Place/Transition Petri nets
  - additionally: unified (black) tokens and constant weights
  - clear mathematical description, possible analysis of properties



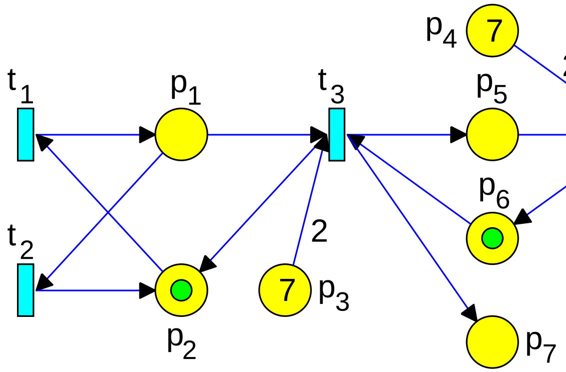
Mathematical description

LMSC
LAMS

- Place/Transition Petri nets
  - net structure (places, transitions, arcs, weights)
    $N = (P, T, A, W)$
    - $P = \{p1, p2, \ldots, p_k\},\ k > 0$
    - $T = \{t1, t2, \ldots, t_l\}, l > 0$
      ($P \cup T = \emptyset$ in $P \cap T = \emptyset$)
    - $A \subseteq (P \times T) \cup (T \times P)$
    - $W: (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$



  - Petri net system $S = (N, m_0)$
    - $m: P \rightarrow \mathbb{N}$ marking (tokens in $p \in P$)
      m can be represented by an integer vector $\boldsymbol{m}$
      m_0 is initial marking

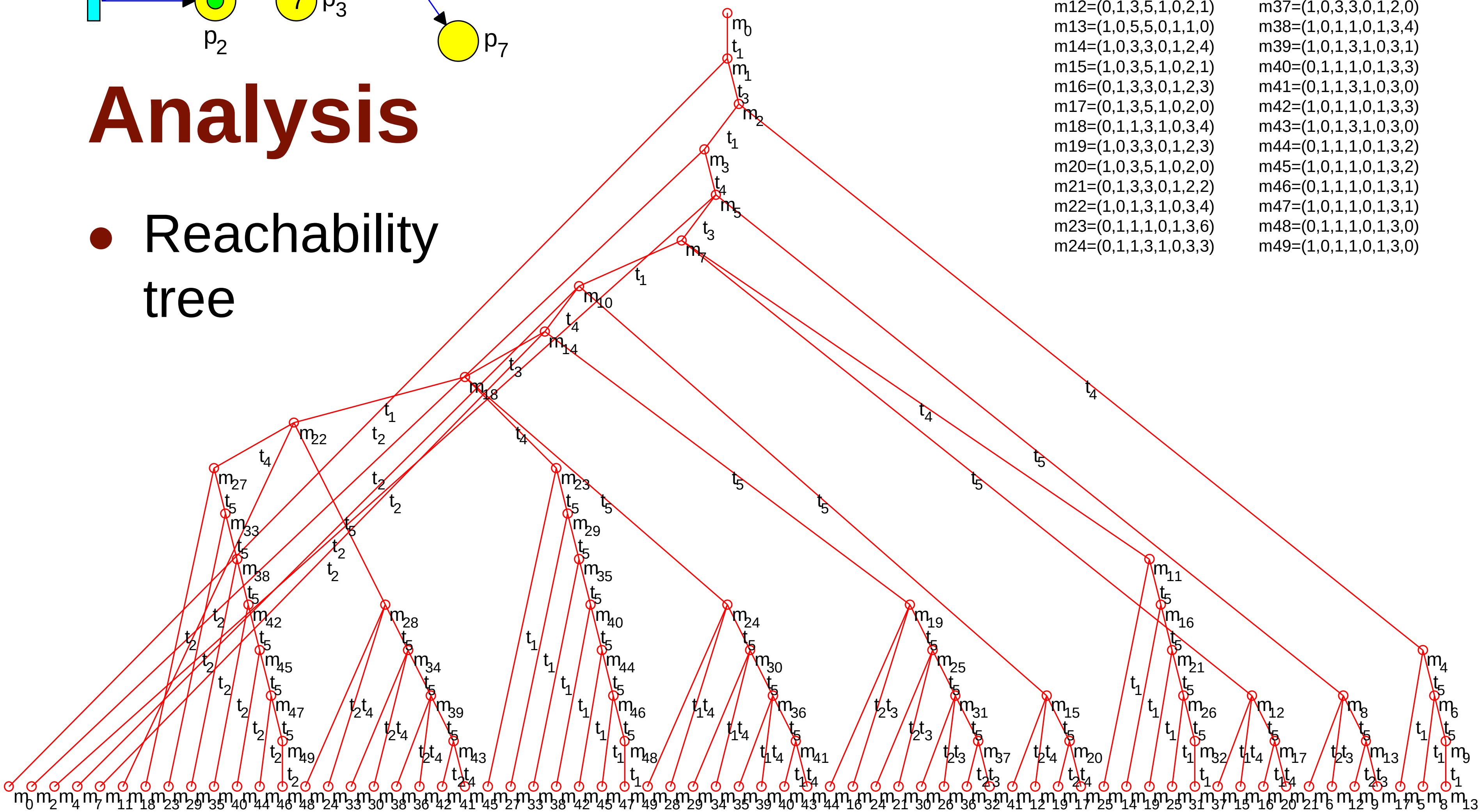


Analysis

- Reachability tree

m0=(0,1,7,7,0,1,0,0)
m1=(1,0,7,7,0,1,0,0)
m2=(0,1,5,7,1,0,1,0)
m3=(1,0,5,7,1,0,1,0)
m4=(0,1,5,5,0,1,1,2)
m5=(1,0,5,5,0,1,1,2)
m6=(0,1,5,5,0,1,1,1)
m7=(0,1,3,5,1,0,2,2)
m8=(1,0,5,5,0,1,1,1)
m9=(0,1,5,5,0,1,1,0)
m10=(1,0,3,5,1,0,2,2)
m11=(0,1,3,3,0,1,2,4)
m12=(0,1,3,5,1,0,2,1)
m13=(1,0,5,5,0,1,1,0)
m14=(1,0,3,3,0,1,2,4)
m15=(1,0,3,5,1,0,2,1)
m16=(0,1,3,3,0,1,2,3)
m17=(0,1,3,5,1,0,2,0)
m18=(0,1,1,3,1,0,3,4)
m19=(1,0,3,3,0,1,2,3)
m20=(1,0,3,5,1,0,2,0)
m21=(0,1,3,3,0,1,2,2)
m22=(1,0,1,3,1,0,3,4)
m23=(0,1,1,1,0,1,3,6)
m24=(0,1,1,3,1,0,3,3)

m25=(1,0,3,3,0,1,2,2)
m26=(0,1,3,3,0,1,2,1)
m27=(1,0,1,1,0,1,3,6)
m28=(1,0,1,3,1,0,3,3)
m29=(0,1,1,1,0,1,3,5)
m30=(0,1,1,3,1,0,3,2)
m31=(1,0,3,3,0,1,2,1)
m32=(0,1,3,3,0,1,2,0)
m33=(1,0,1,1,0,1,3,5)
m34=(1,0,1,3,1,0,3,2)
m35=(0,1,1,1,0,1,3,4)
m36=(0,1,1,3,1,0,3,1)
m37=(1,0,3,3,0,1,2,0)
m38=(1,0,1,1,0,1,3,4)
m39=(1,0,1,3,1,0,3,1)
m40=(0,1,1,1,0,1,3,3)
m41=(0,1,1,3,1,0,3,0)
m42=(1,0,1,1,0,1,3,3)
m43=(1,0,1,3,1,0,3,0)
m44=(0,1,1,1,0,1,3,2)
m45=(1,0,1,1,0,1,3,2)
m46=(0,1,1,1,0,1,3,1)
m47=(1,0,1,1,0,1,3,1)
m48=(0,1,1,1,0,1,3,0)
m49=(1,0,1,1,0,1,3,0)



Matrix description

LMSC
LAMS

- Transition input matrix $\boldsymbol{Pre}$
  - element in i-th row and j-th column describes the arc

    from p_i to t_j: $$a_{ij}^{Pre} = \begin{cases} W(p_i, t_j), & (p_i, t_j) \in A \\ 0, & (p_i, t_j) \notin A \end{cases}$$

- Transition output matrix $\boldsymbol{Post}$
  - element in i-th row and j-th column describes the arc

    from t_j to p_i: $$a_{ij}^{Post} = \begin{cases} W(t_j, p_i), & (t_j, p_i) \in A \\ 0, & (t_j, p_i) \notin A \end{cases}$$

- Incidence matrix: $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$

$$(p_i, t_j) \in A \wedge (t_j, p_i) \in A \Rightarrow c_{ij} = W(t_j, p_i) - W(p_i, t_j)$$

Algebraic analysis

- Firing vector $\boldsymbol{u} = [0, \dots, 0,1,0, \dots, 0]^T$
  - 1 at the j-th component represents firing of transition t_j
- Transition t_j is enabled if $\boldsymbol{m} \geq \boldsymbol{Pre} \cdot \boldsymbol{u}$
- State equation

$$\boldsymbol{m}' = \boldsymbol{m} + \boldsymbol{C} \cdot \boldsymbol{u}$$

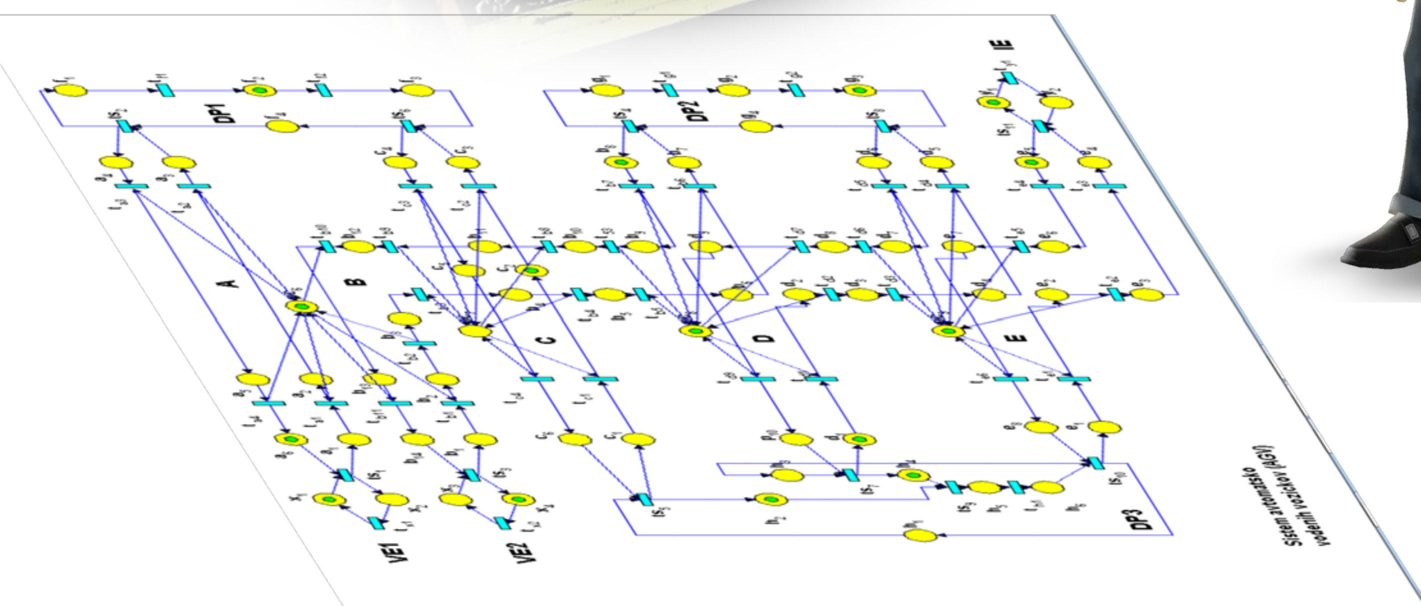
- A necessary condition for reachability of a marking
  - a marking $\boldsymbol{m}$ is only reachable from $\boldsymbol{m}_0$ if

$$\boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{x} = \boldsymbol{m}$$

has an integer solution $\boldsymbol{x}$

Applicability in industrial informatics?

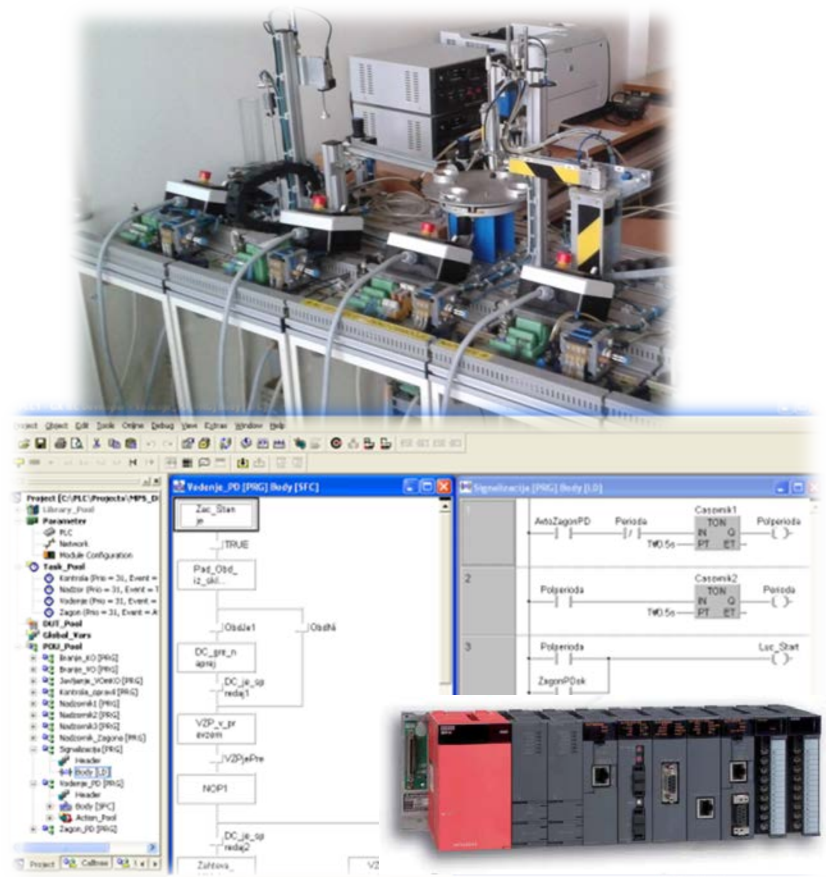
LMSC
LAMS



Application domains

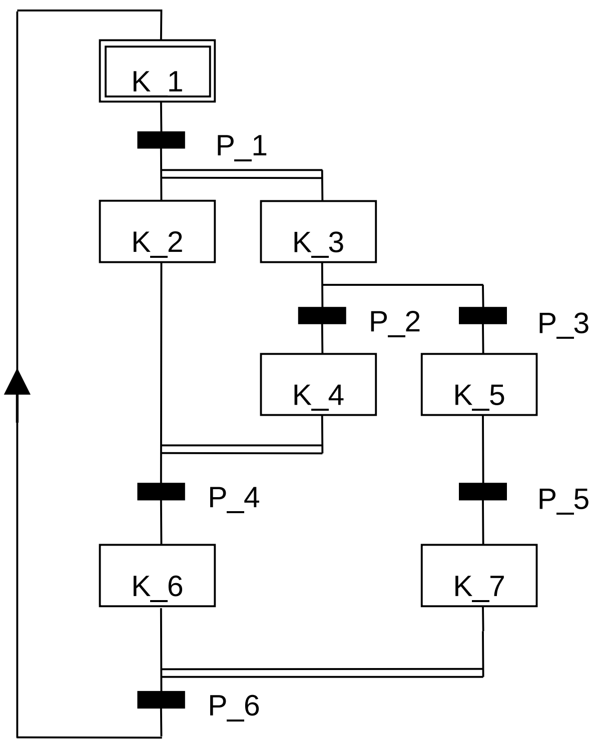
LMSC
LAMS

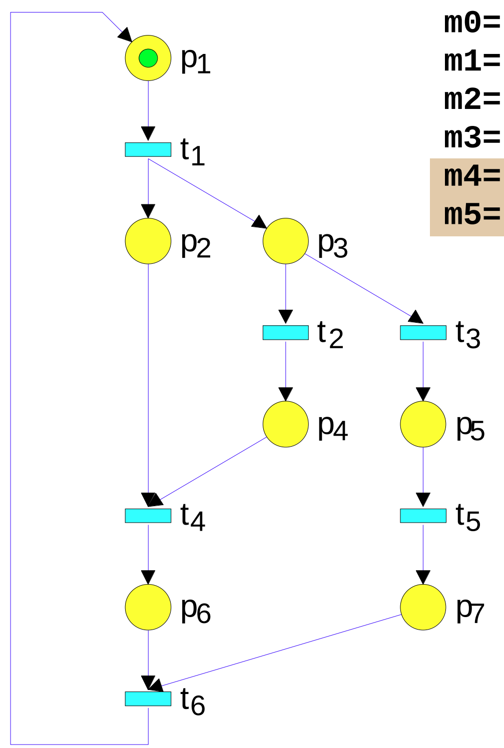
- analysis of manufacturing procedures and processes
- structural optimization of production lines
- specification of recipes in batch systems
- → control logic design
- human-machine interface design
- resource allocation
- → operations scheduling
- modelling of decision rules …



Control logic design

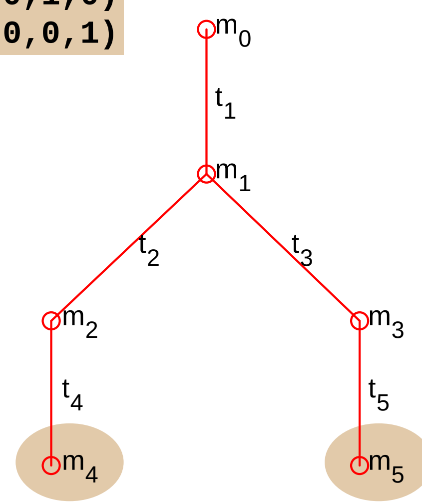
- Verification





Reachable markings analysis:

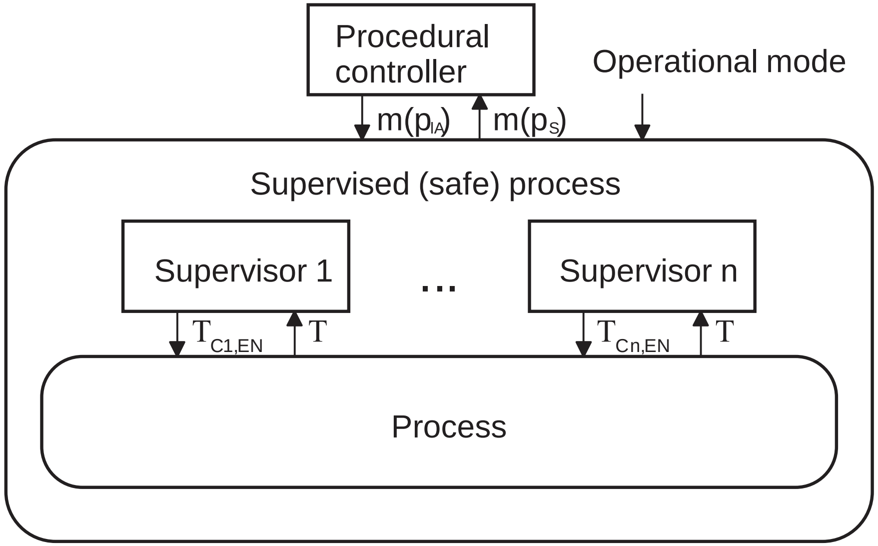
```
m0=(1,0,0,0,0,0,0)
m1=(0,1,1,0,0,0,0)
m2=(0,1,0,1,0,0,0)
m3=(0,1,0,0,1,0,0)
m4=(0,0,0,0,0,1,0)
m5=(0,1,0,0,0,0,1)
```



Markings m_4 and m_5 are blocking (dead markings) - deadlock

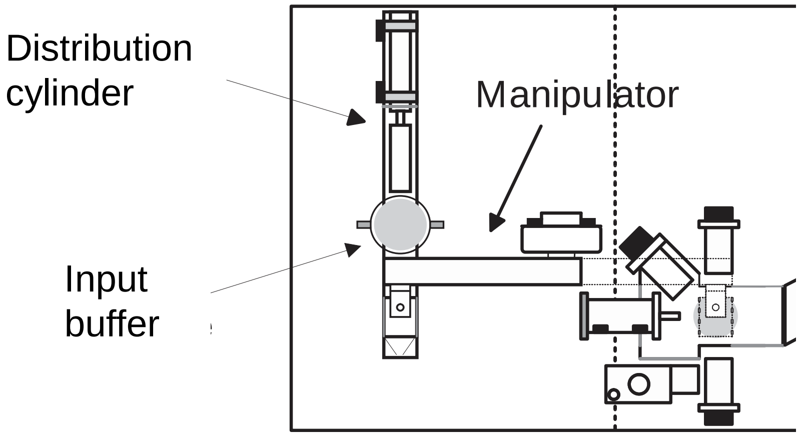
Control logic design

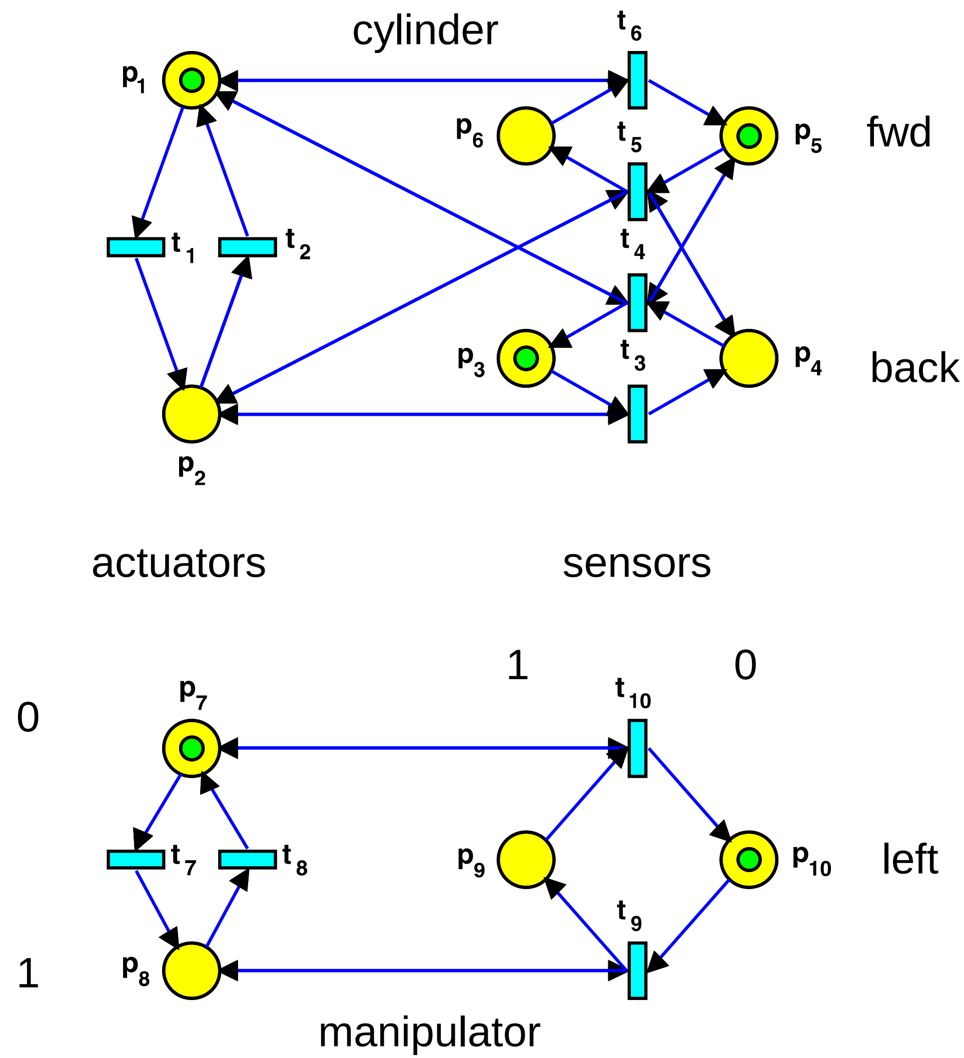
- Combined synthesis verification approach
  - safety measures (interlocks)
    - supervisory control synthesis
  - operational procedures
    - overall system model is built
    - verification
  - this way control system properties can be analysed, which cannot be addressed by analysis of the process model or control logic model alone



G. Mušič, D. Matko. An admissible-behaviour-based analysis of the deadlock in Petri-net controllers. Simulation modelling practice and theory, 16 (8): 1077-1090, 2008.

Example – part of production line





- Collisions between cylinder and manipulator should be prevented:

$$m(p_2) + m(p_8) \leq 1$$

Example – part of production line

- Supervisor synthesis

$$\boldsymbol{L} = [0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0]$$

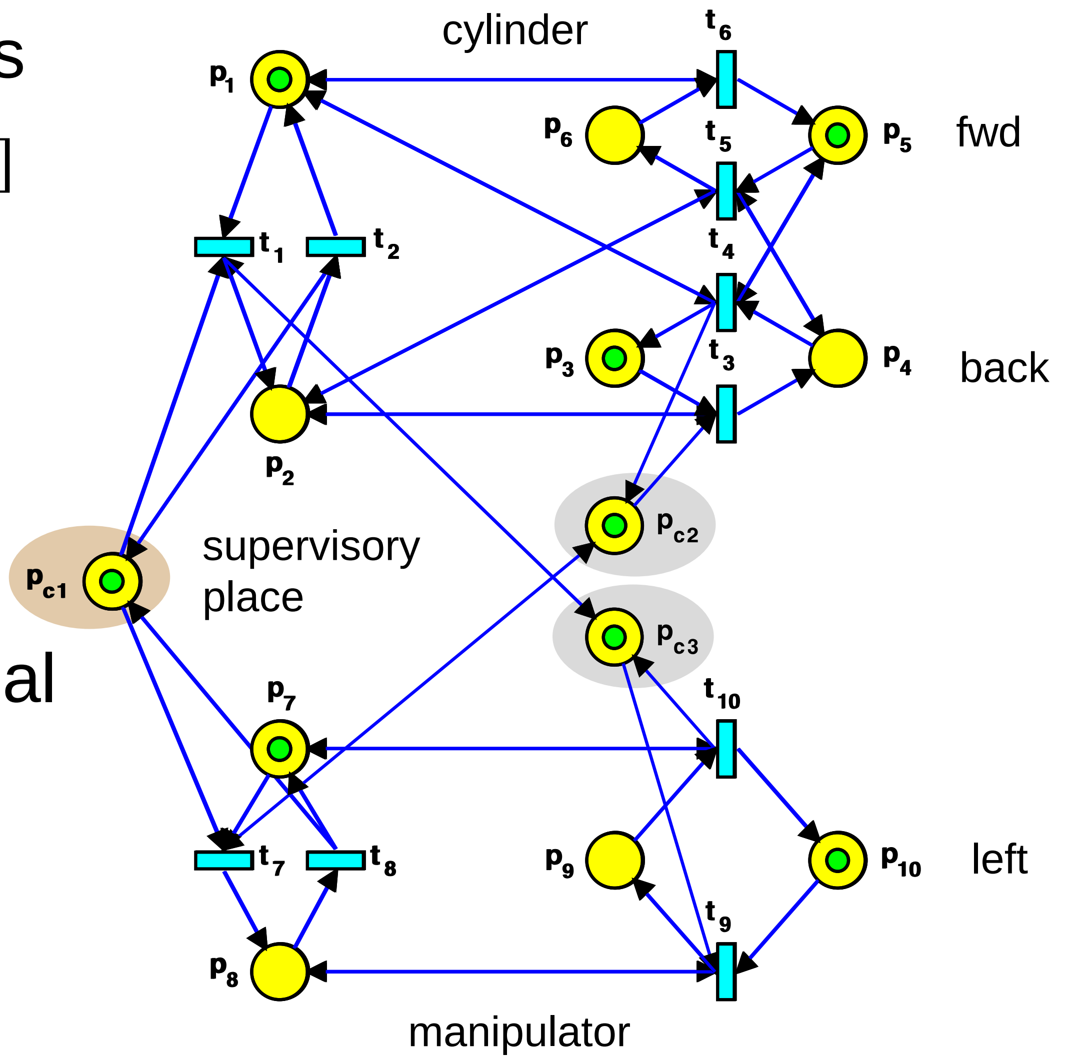
$$\mathbf{b} = 1$$

$$\boldsymbol{C}_c = -\boldsymbol{LC}$$

$$= [-1\ 1\ 0\ 0\ 0\ 0\ \ -1\ 1]$$

$$\boldsymbol{m}_{c0} = \boldsymbol{b} - \boldsymbol{L}\boldsymbol{m}_0 = 1$$

- With additional constraints, additional supervisory places are calculated

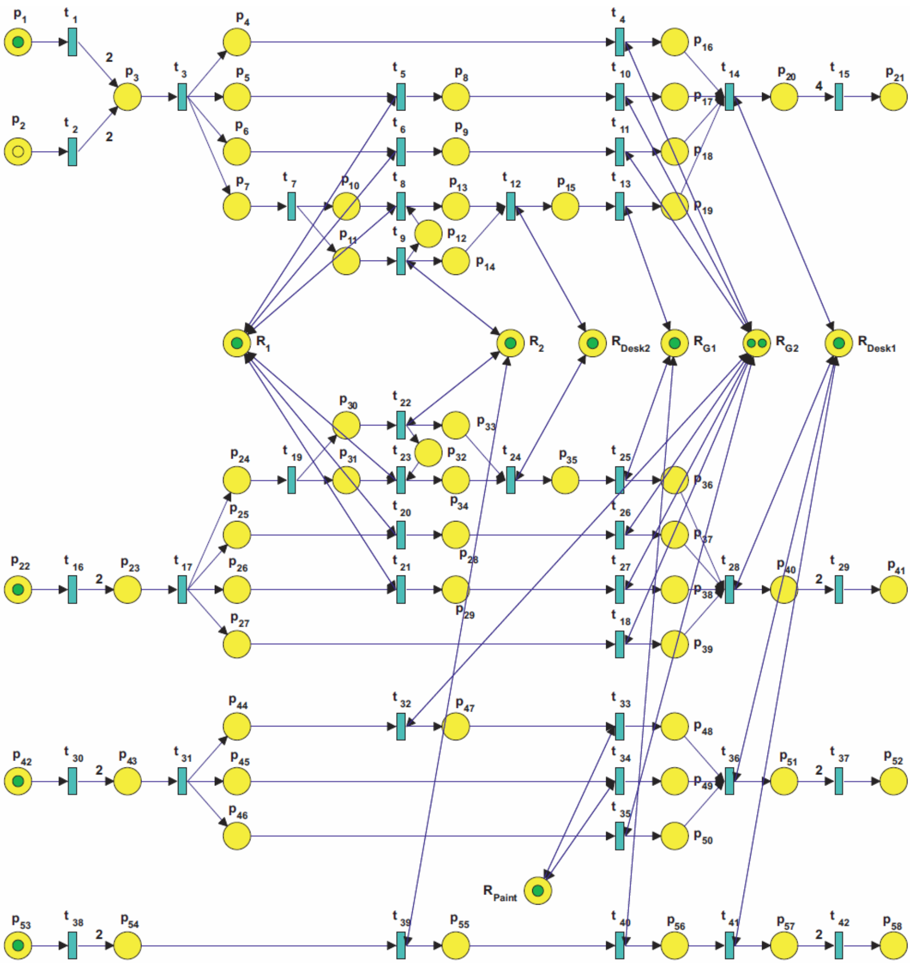


Operations scheduling

source: KBA Group Newsletter

Petri nets and operations scheduling – modelling

- Development of model-building algorithms for scheduling problems



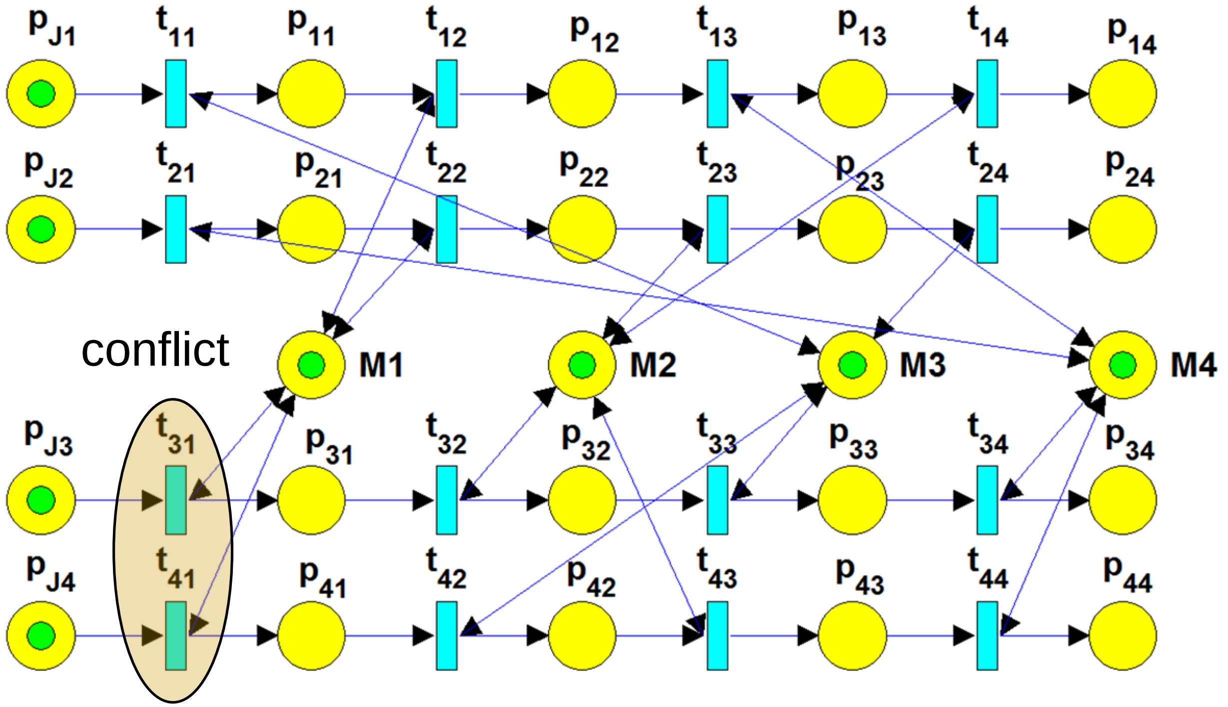
G. Mušič, T. Löscher, D. Gradišar. An open Petri net modelling and analysis environment in Matlab. International Mediterranean Modelling Multiconference 2006, Barcelona, Spain.

D. Gradišar, G. Mušič. Production-process modelling based on production-management data: a Petri-net approach. International Journal of Computer Integrated Manufacturing, 20 (8): 794-810, 2007.

D. Gradišar, G. Mušič. Petri-net modelling for batch production. IFAC Conference on Manufacturing Modelling, Management and Control, 2013, Saint Petersburg, Russia.

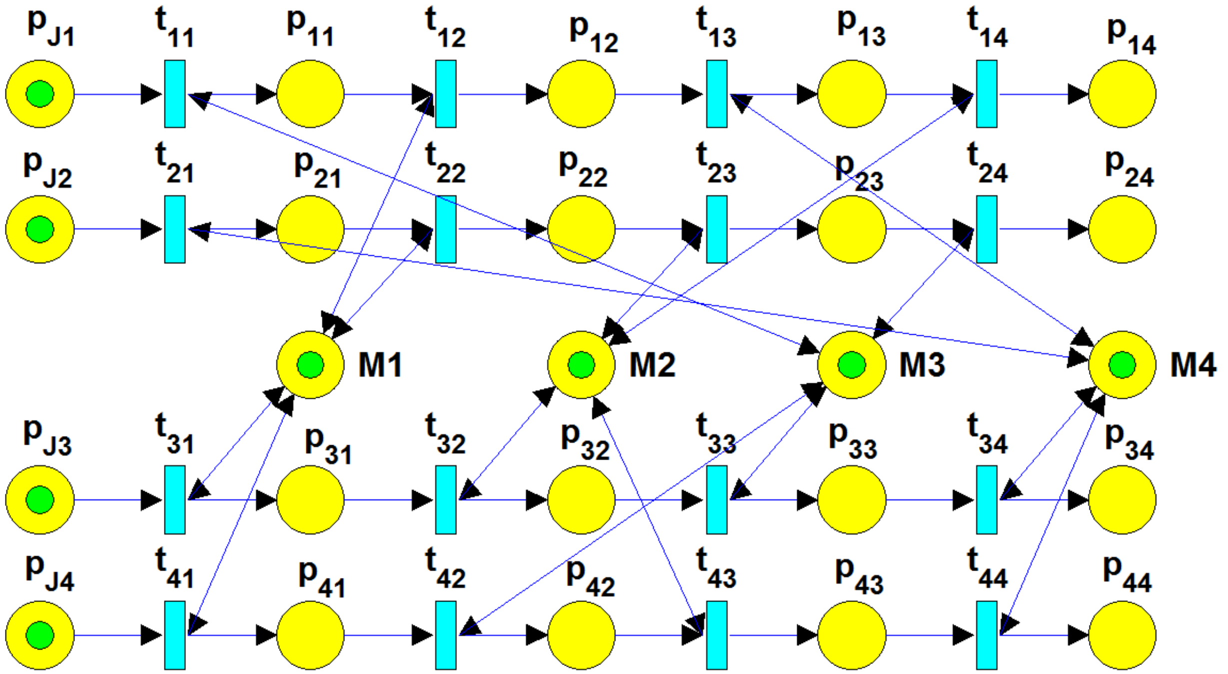
Petri nets and operations scheduling – optimization

- Based on the net simulation
  - operations durations must be considered – timed simulation
  - on the fly conflict resolution – design of a schedule

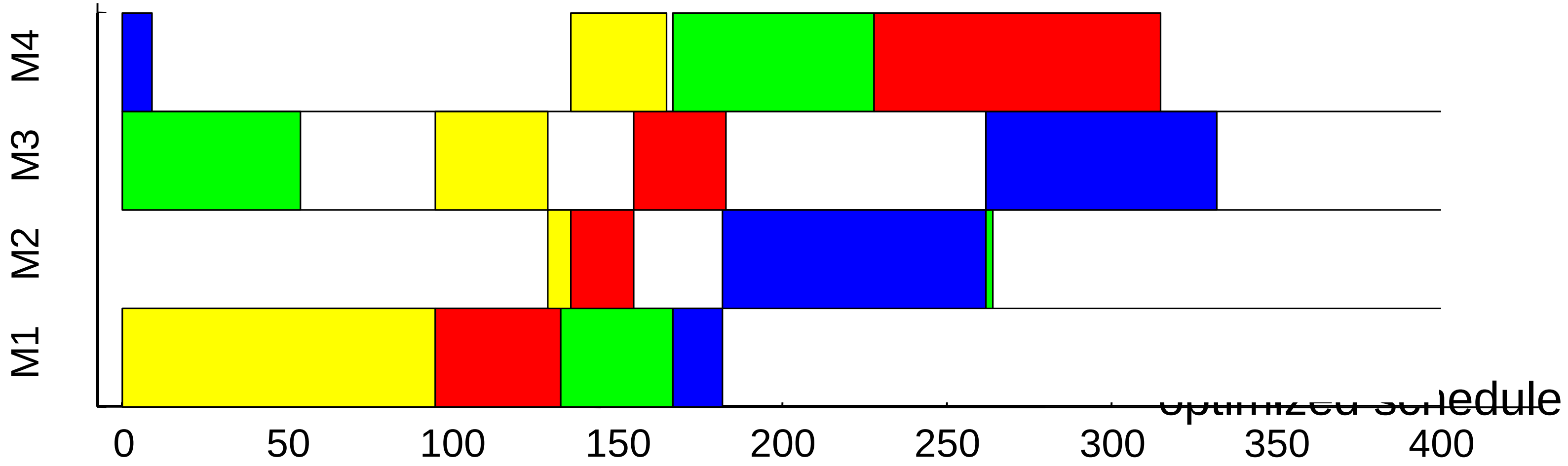


Petri nets and operations scheduling – optimization

- Based on the net simulation
  - operations durations must be considered – timed simulation
  - on the fly conflict resolution – design of a schedule



Scheduling example

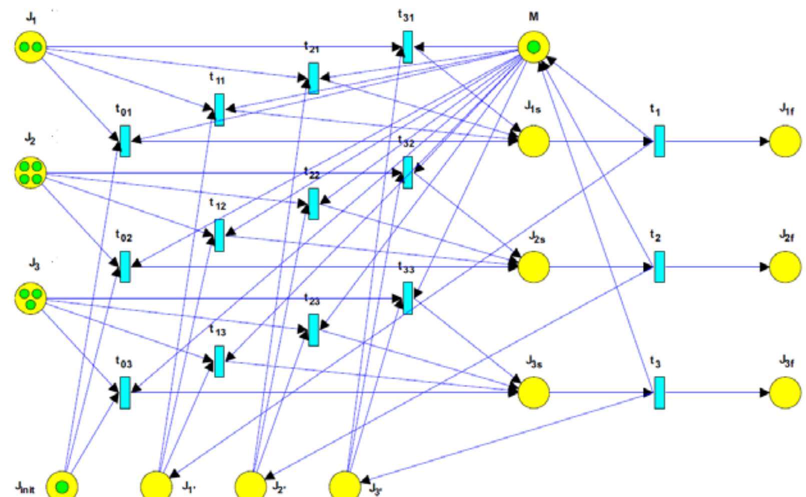


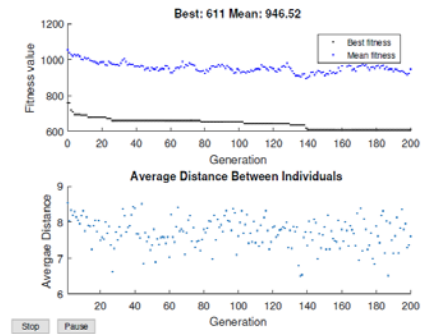
- Importance of a proper net state space exploration
  - consideration of possibility that an enabled transition is not immediately triggered

M. A. Piera, G. Mušič. Coloured Petri net scheduling models : timed state space exploration shortages. Transactions of IMACS, Mathematics and computers in simulation, 82 (3): 428-441, 2011.

More complex scheduling problems

- Problems with sequence dependent setup times
- Local search algorithms
  - solution space S; search in a »neighbourhood« N of a given solution $s \in S$, $N : S \to 2^S$
    - simulated annealing (SA), threshold accepting (TA), tabu search (TS), genetic algorithms (GA)
  - determination of solution quality – PN simulation
- Simulation-based scheduling approach





G. Mušič. Simulation-optimization of schedules with sequence dependent setup times based on Petri net models. The 28th European Modeling & Simulation Symposium, September 26 - 28 2016, Cyprus.

Usefulness of Petri nets in operations scheduling

LMSC
LAMS

- Larger generalization capability in comparison to other formalisms
  - many real scheduling problems can be modelled in the same way as standard problems
  - unified treatment of typical situations in automated production:
    - tasks synchronization, shared resources, material flow joining and splitting, maintenance of process operation and deadlock avoidance …
- The same model can be used to
  - analyse system properties
  - simulate operation scenarios
  - use various schedule optimization methods

Wider importance of Petri nets

LMSC
LAMS

- Nets as a universal model
  - revival of the »space computer« idea
    - initially K. Zuse, Rechnender Raum, 1969
  - C. A. Petri, Das Universum als großes Netz, Ist das Universum ein Computer?, Spektrum der Wissenschaft, Spezial, 3/2007
    - at the Planck length quantum computing can be described by digital systems using combinatorial models
    - the universe can be studied using discrete nets
    - discrete systems would suffice to explain even quantum and relativistic phenomena