Modelling an Industrial Transportation Facility Using Coloured Petri Nets

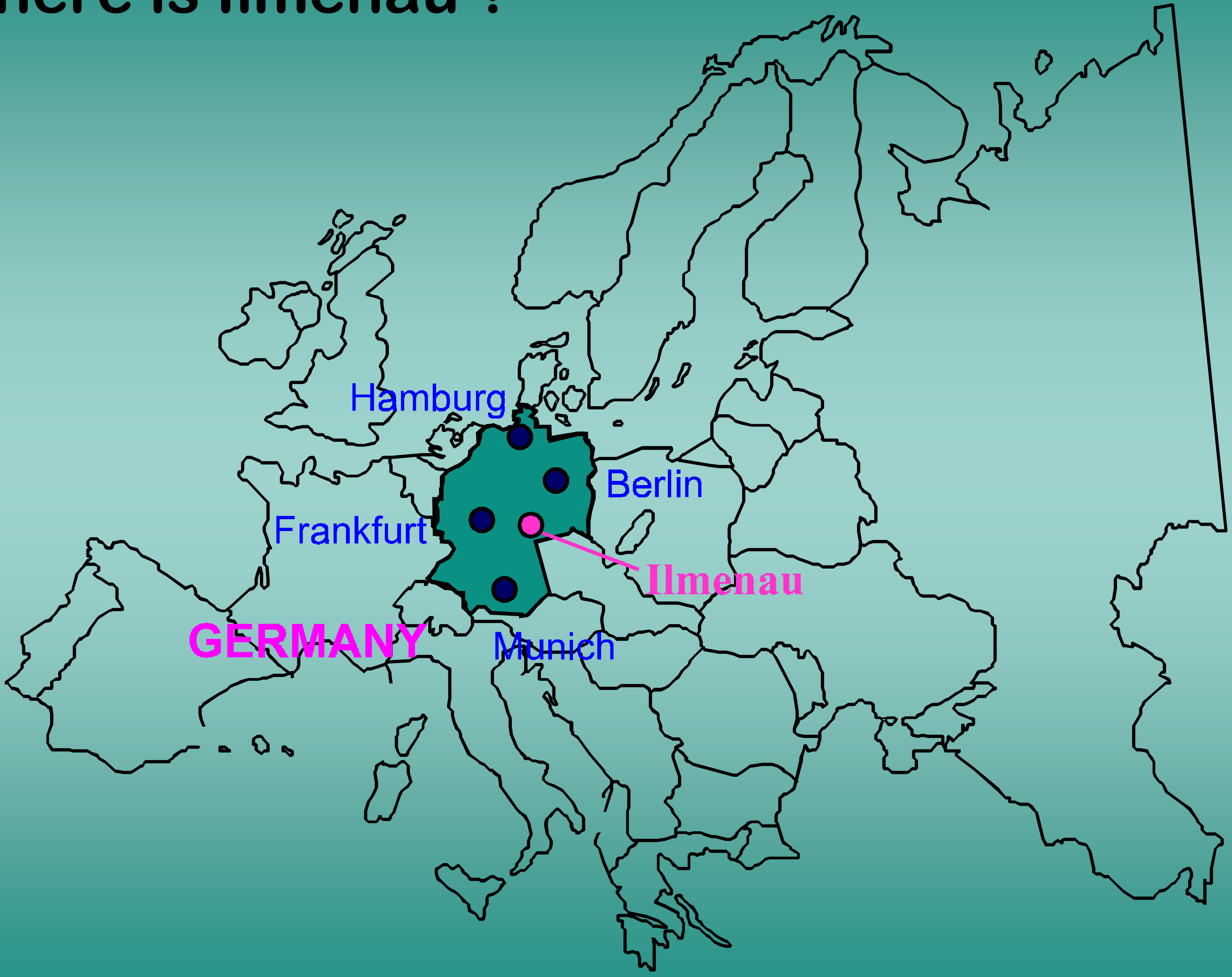
Bernd Däne, Angela Mölders

Andrea Melber, Wolfgang Fengler

Technical University of Ilmenau
Faculty of Informatics and Automation
Institute of Theoretical and Technical Computer Science

TU Ilmenau

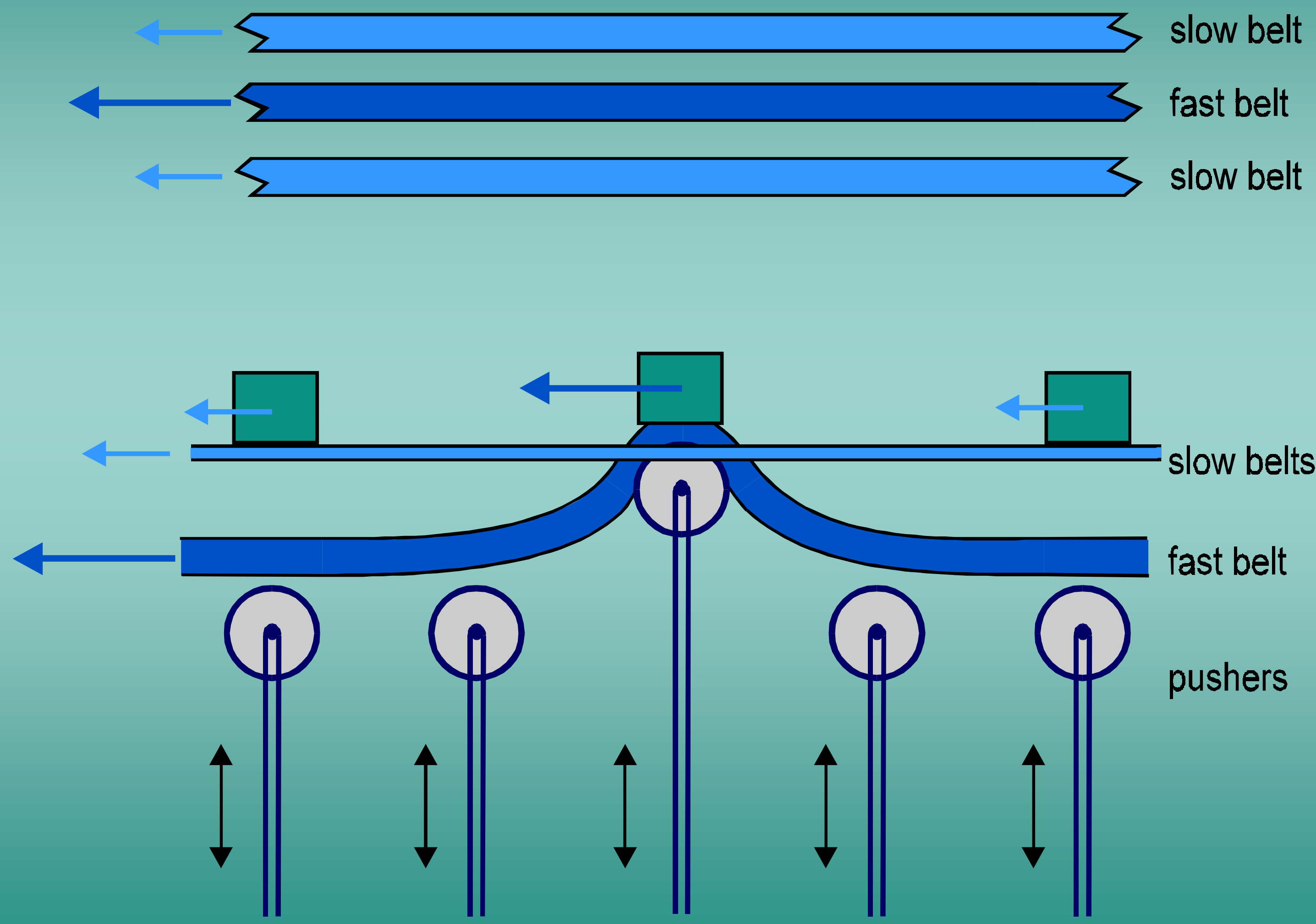
Where is Ilmenau ?



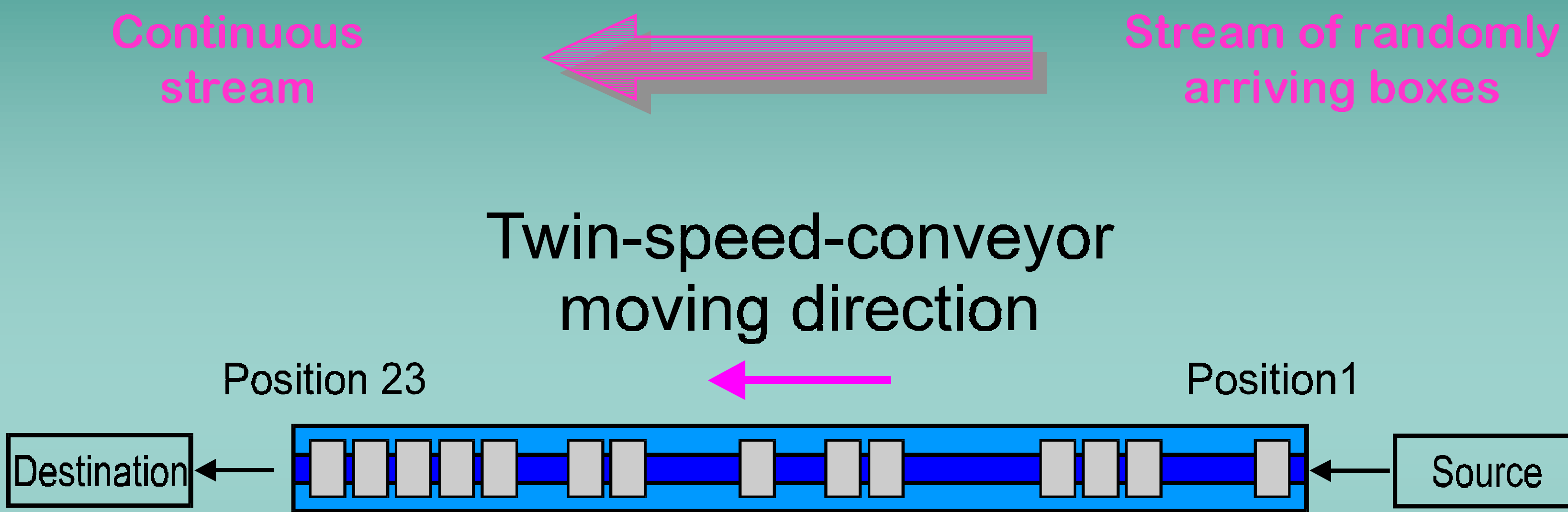
Topics

1. The Twin Speed Conveyor
2. Modelling the Example Using Coloured Petri Nets
3. Evaluating the Model
4. Implementing Target Software

Twin Speed Conveyor



Abstraction of the Process



time-discrete approximation

position_length = length_of_box + minimum_distance

inner belt's speed: one position per clock

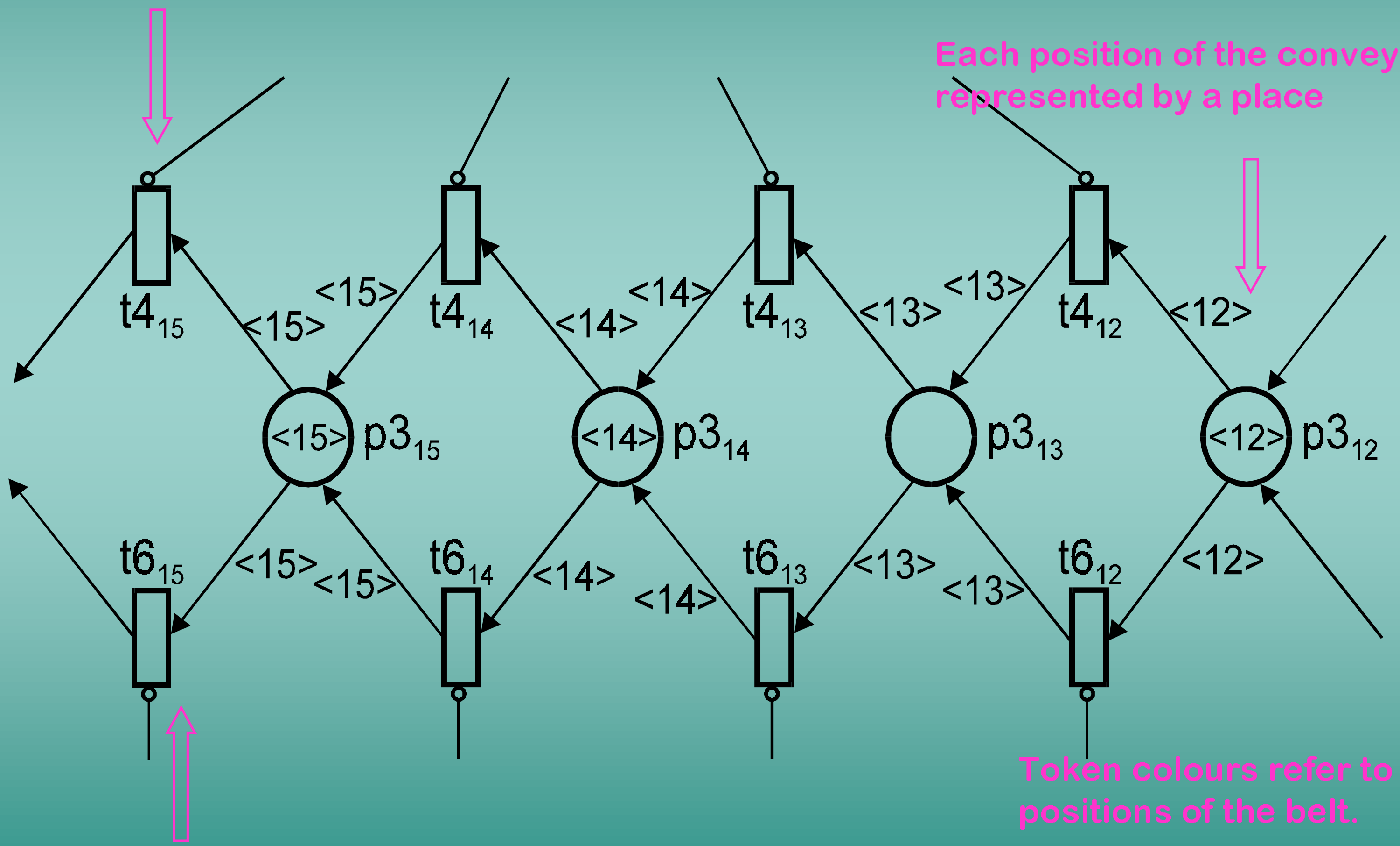
outer belts' speed: one position per 2,4,6,9 or 14 clocks

Kernel of the Petri Net Model

Normal movement of boxes with various velocities

All of these transitions are activated simultaneously.

Each position of the conveyor is represented by a place



Token colours refer to the positions of the belt.

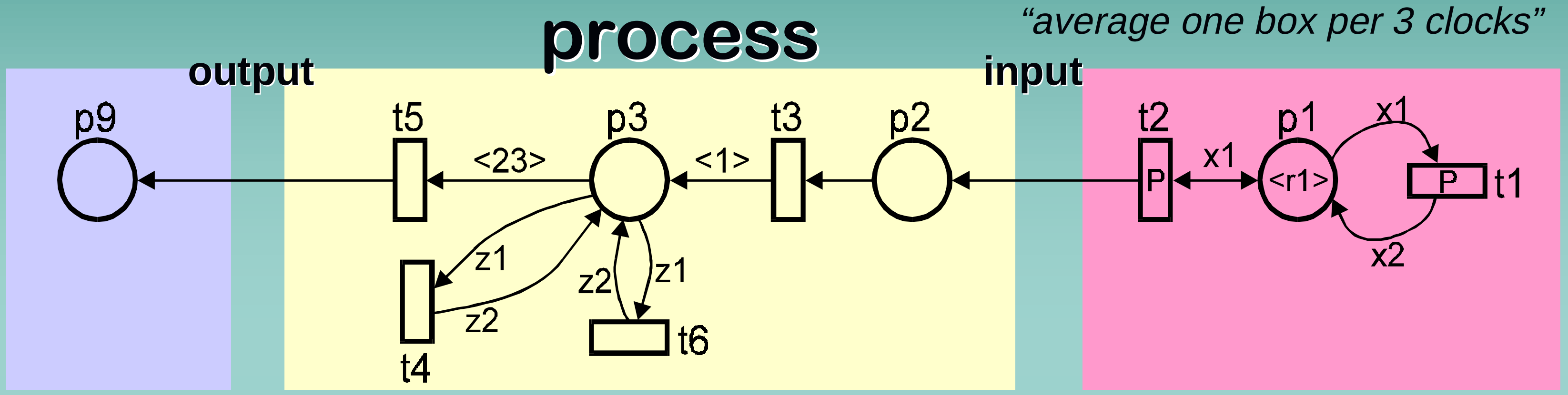
Movement of boxes controlled by pushers

Individual activation of these transitions is possible.

TU Ilmenau

Process Model

Boxes arrive at random distances but at a given rate:
"average one box per 10 clocks"
"average one box per 3 clocks"



- Positions of the belt are encoded into different token colours:
  <1>, <2>, ... <23>
- Symbols z1 and z2 realize the boxes' movement by recolouring the token colours
- t4 represents the normal movement on the outer belts
- t6 represents accelerated movement caused by pushers
- t5 and t3 adapt the process to the environmental requirements

Design of the Process Control Component

Problems:

- Observation of the boxes on the conveyor belt
- Pushers' controlling
- Adapting the outer belts' velocity to the number of boxes on the belt

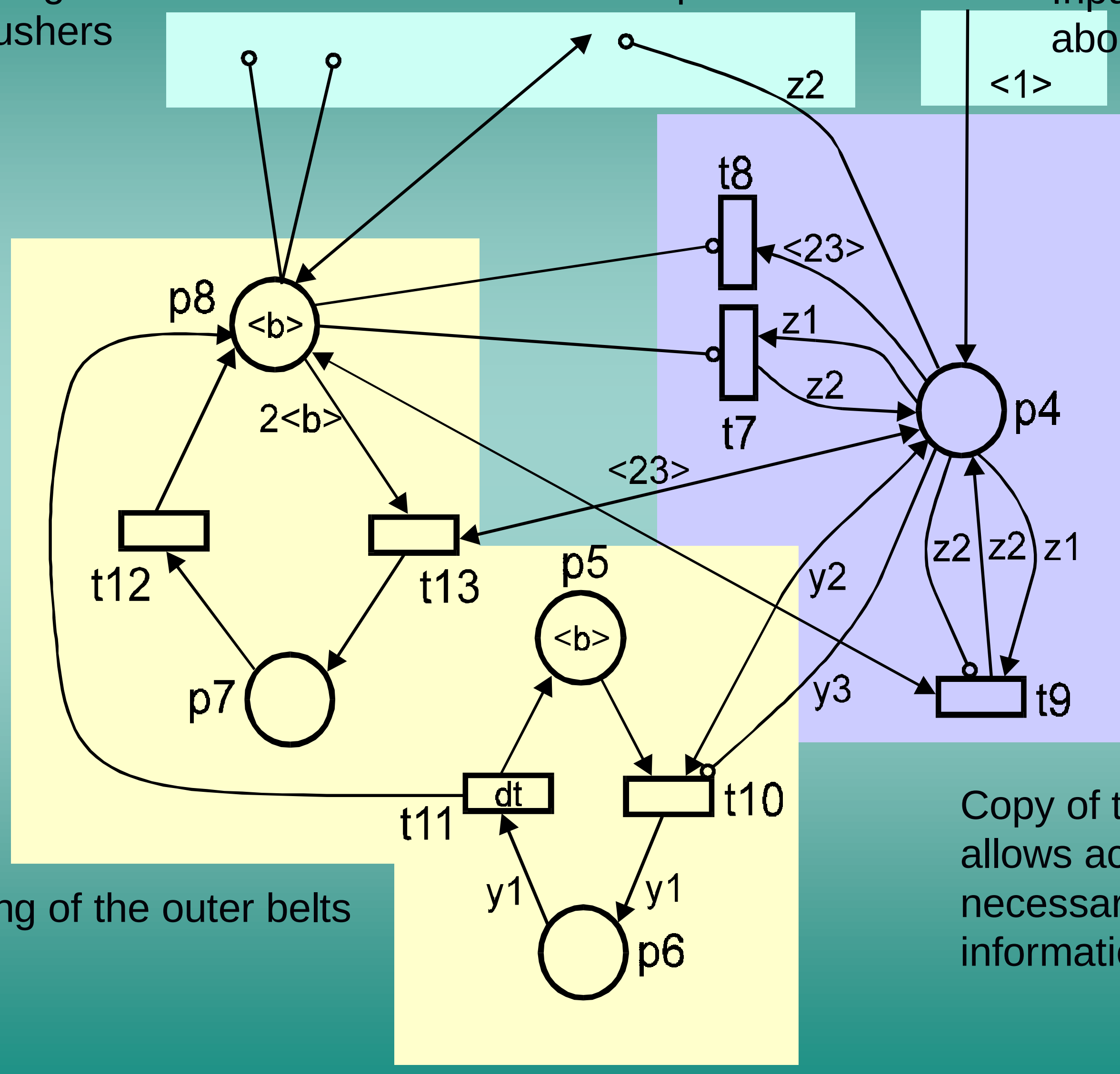
Solution:

- Activation of pushers according to the rule:
  pusher_i_on
  ***if** box_at_position_i **and not** box_on_position_i+1*
- Outer belts' speed depends on the number of boxes in queue starting at position 23:
  The more boxes are in queue the higher the velocity will be.

Control Algorithm

Outgoing control information for the belts' speed and the pushers

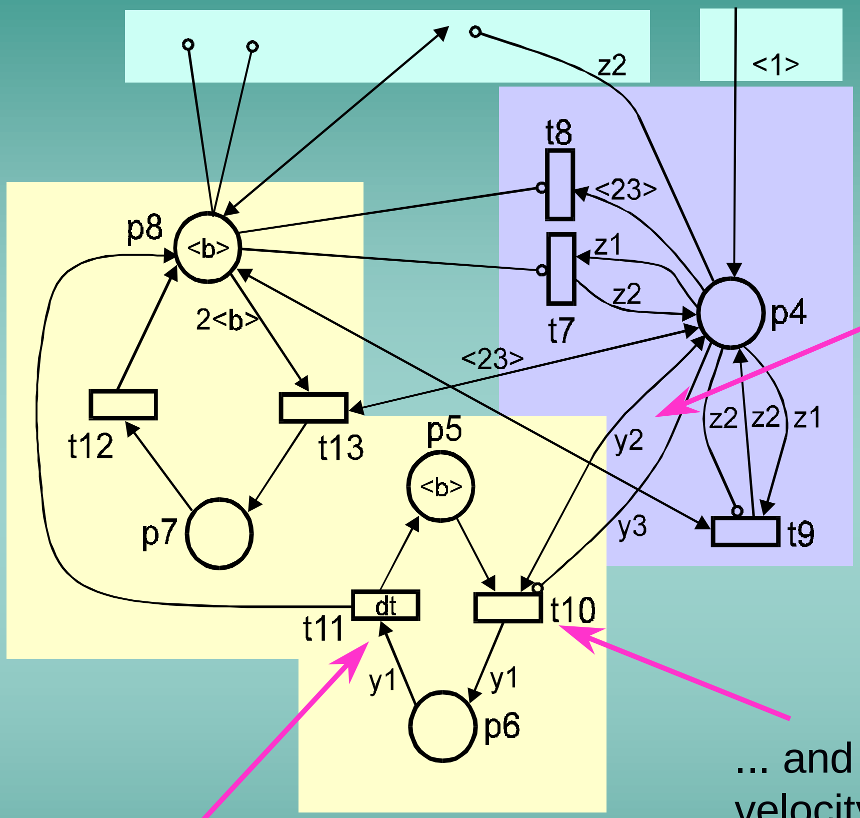
Input information about incoming boxes



Timing of the outer belts

Copy of the process model allows access to all necessary process state information

Encoding of Various Velocities of Outer Belts



Arcs with symbols detect how many boxes are in queue e.g. if there are 3 or 4 boxes in queue the value of the symbols expands to:

y2: <21>and <22>and<23>

y3: <19 >and<20>

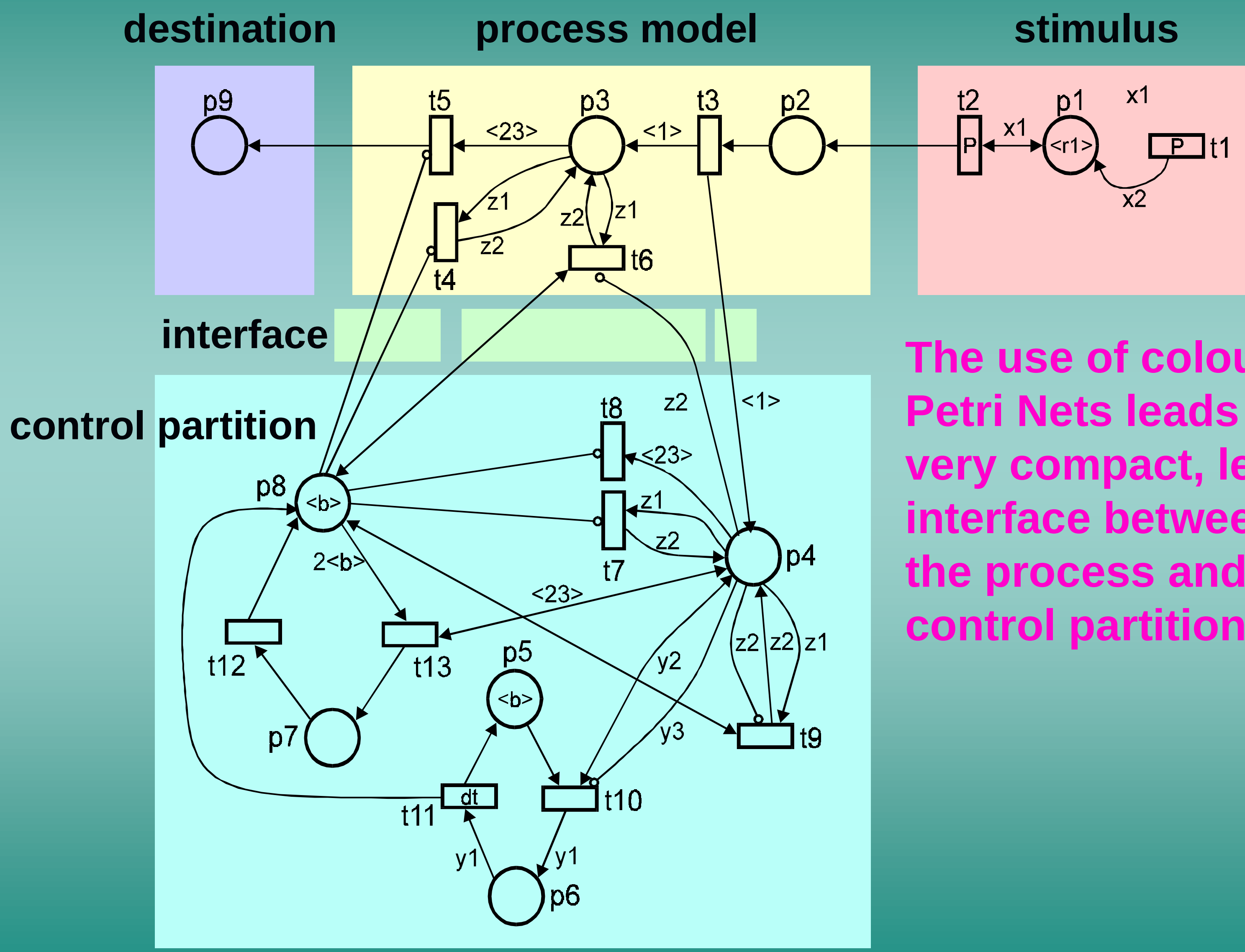
This corresponds to velocity v2

... and firing of t10 encodes the required velocity into the token colour of p6:

<v1>,..., <v5>

Depending on the token colour in p6 firing of t11 takes a different number of clock ticks

Complete Petri Net



The use of coloured Petri Nets leads to a very compact, lean interface between the process and the control partition

Evaluation of the Model

... by formal analyzing

- The control partition of the net has to be free of conflicts.
- In the unfolded net the places representing the belt have to be 1-safe.
- Invariant analysis can be done.
- Unmeant deadlocks can be detected.

... by simulating

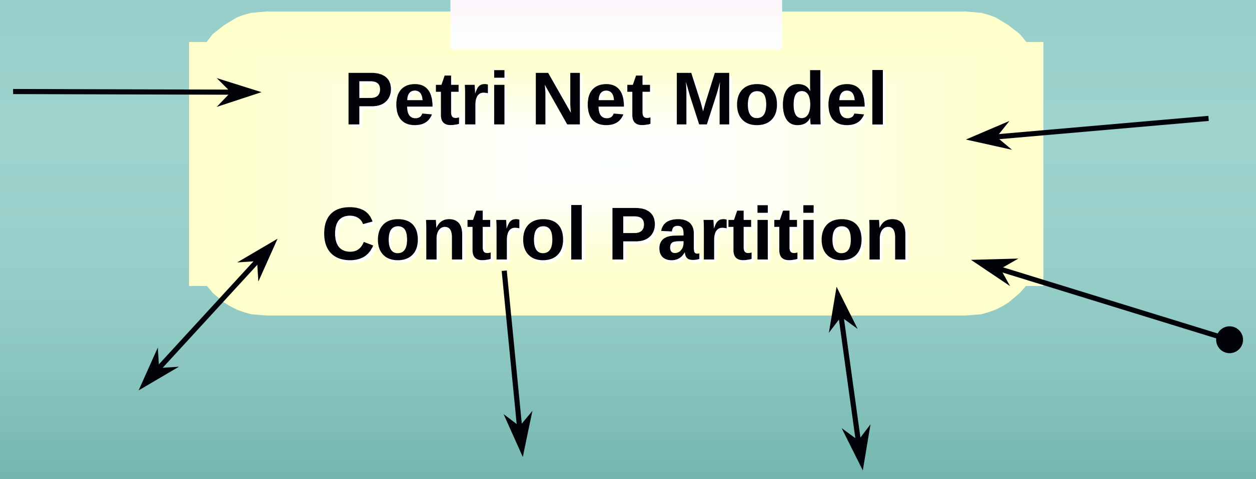
- Effects of changing some net parameters are immediately visible.
- Fixing the input rate to maximum or minimum shows worst case behaviour of the model.
- Statistical evaluation can be used for prediction of quantitative properties of the process.

Implementing Target Software

Implementation of target systems

manual interaction is required

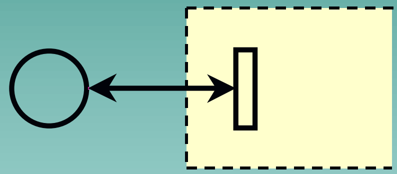
tools for target languages c/c++ and APL-2



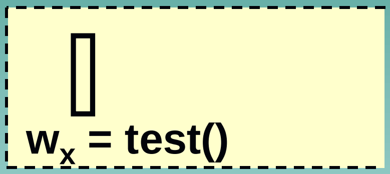
The arcs cutted by the border of control partition have to be transformed into input or output activities of control software.

Different Types of Cutted Arcs

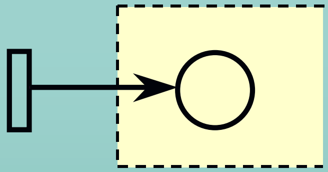
Incoming test-arc



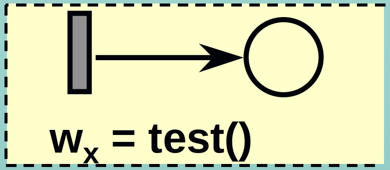
replace by...



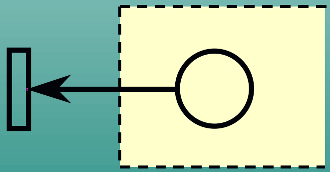
Incoming post-arc (safe)



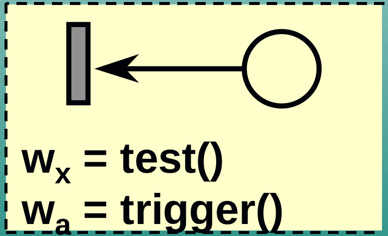
replace by...



Outgoing pre-arc



replace by...



Conclusions

- Discrete-time Coloured Petri Nets have been used for detailed modelling of a complex conveyor transportation process.
- The folded (coloured) form of representation allows a better understanding of the net model compared to the original (unfolded) net.
- Simulation and formal analysis of the model led to the detection of design errors and to a considerable improvement of the system architecture.
- Petri Nets (again) proved to be well suited for supporting (almost) all phases of the development process for industrial control software.

Further Work

- At the price of reduced analyzability, the modelling power of Petri Nets can be increased by introducing additional attributes such as
  - tokens with data structures
  - fuzzy extensions for solving conflicts.
- The tools for simulation and analyzing these nets require further improvement.
- In particular, developing of graphical front-ends will lead to more ergonomic application-specific layers.