Budapest University of Technology and Economics
Faculty of Transportation Engineering and Vehicle Engineering
Department of Control for Transportation and Vehicle Systems

# Formal Verification of Railway Interlocking Systems

## Thesis and current work

Balázs Farkas

# The Aim of Interlocking



- The aim: efficient and safe organization of railway traffic
  - Automated functions
  - Safe design, behaviour
- Given properties of railways:
  - Railway track (permanent way) → need for turnouts
  - Low adhesion (long breaking distances) → need for (approach) signalling
- Trains run on locked routes
  - Signal is allowed to set clear only if all points are set and locked
  - Points are allowed to release and set only if the signal is set to stop
  - Conflicting train routes are not allowed

# Different Approaches to Interlocking Design



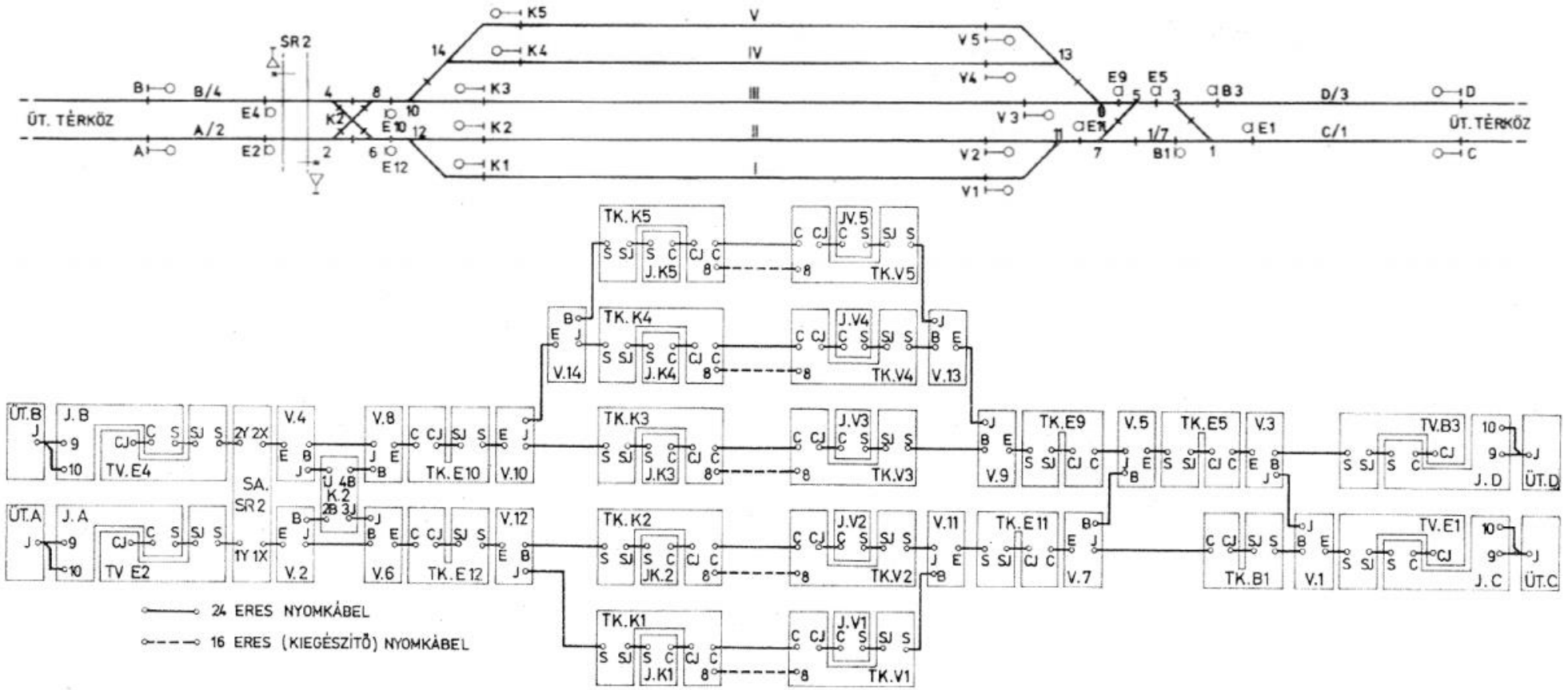
## Geographical principle

- Like a distributed system
- The elements are connected to each other according to their position
- Every element has its own functionality

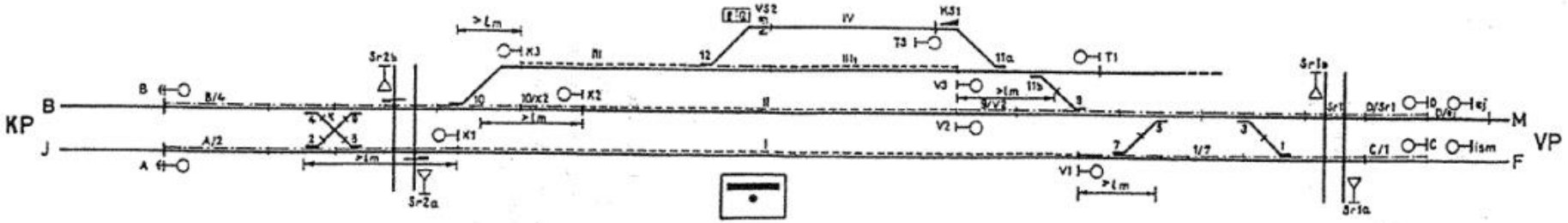
## Tabular principle

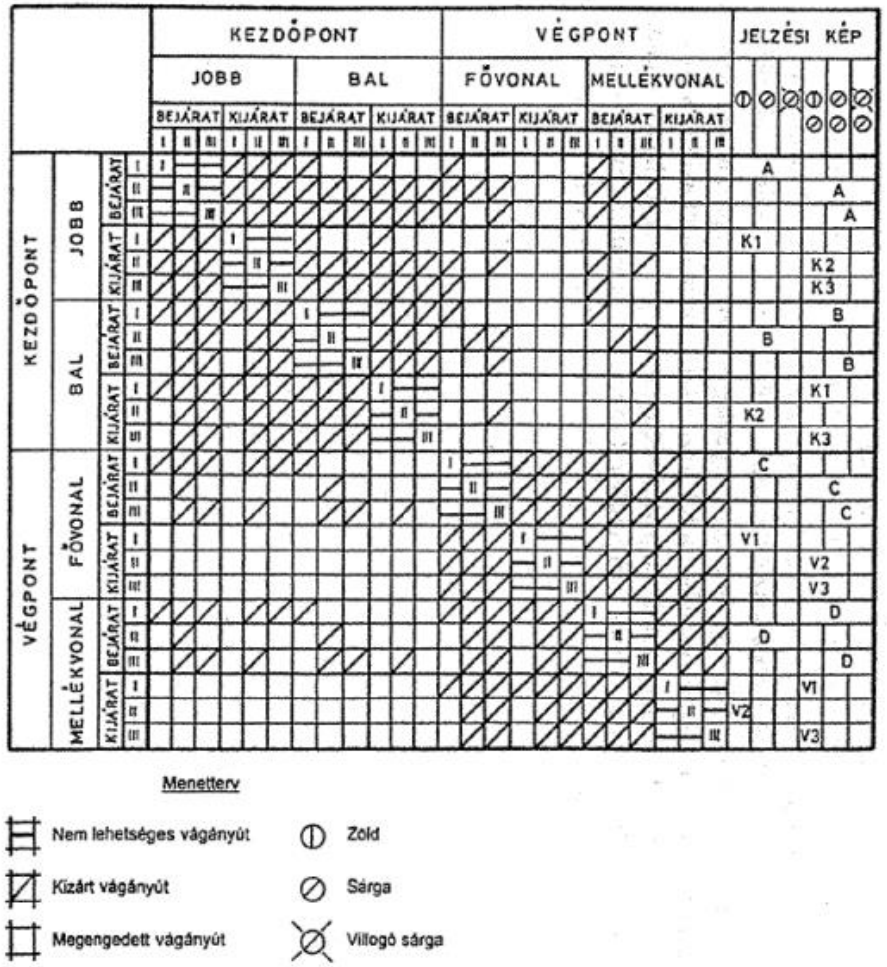
- Like a centralised system
- Every locking connection (between routes or routes and elements) is stored in tables
- The central logic controls the elements based on the tables

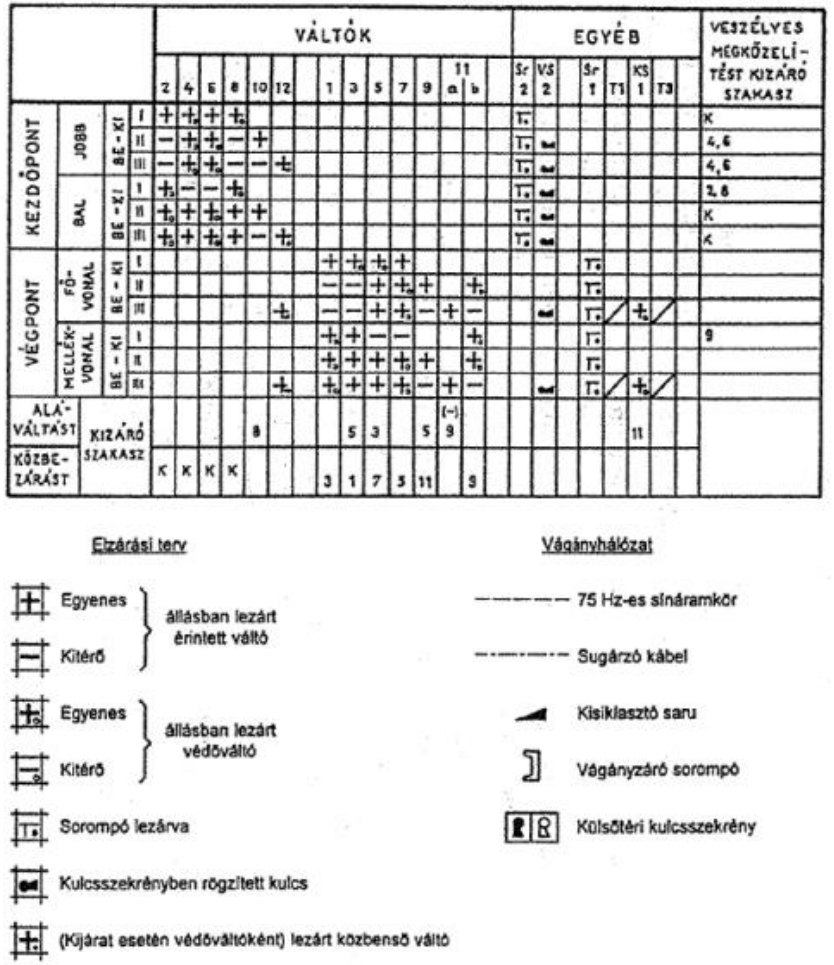
# Example of the geographical principle



# Example of the tabular principle



# Differences between Development & Planning



## Development of Interlocking

- (Interlocking as a product)
- Well-determined process
- With the help of development, verification and validation techniques given by standards
- Verification through the process

## Planning of Interlocking

- (Interlocking as an application)
- Based on some technical prescription, the experiences of the planner and „unwritten law”
- Practicably one-time verification
- Inspection manually

# Thesis Overview – Goals

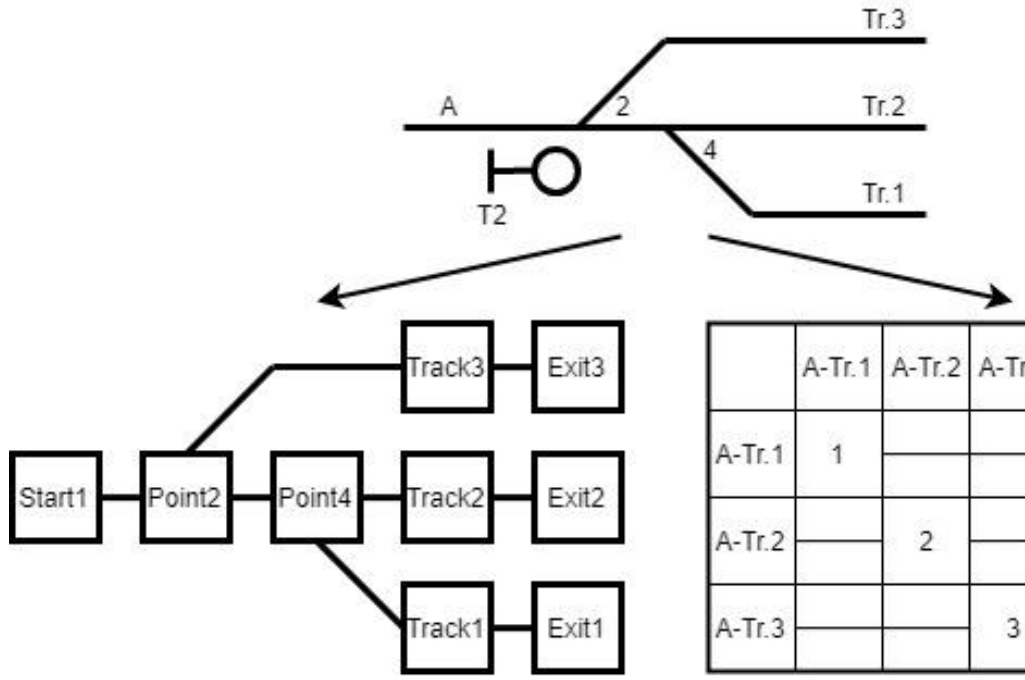


- Modelling the two interlocking design approaches
- Verification via model checking
- Use of two different tools (principles)
  - UPPAAL (finite state machines)
  - PetriDotNet (Petri nets)
- Getting experiences in interlocking modelling
  - Pilot project to further works

# Start Up

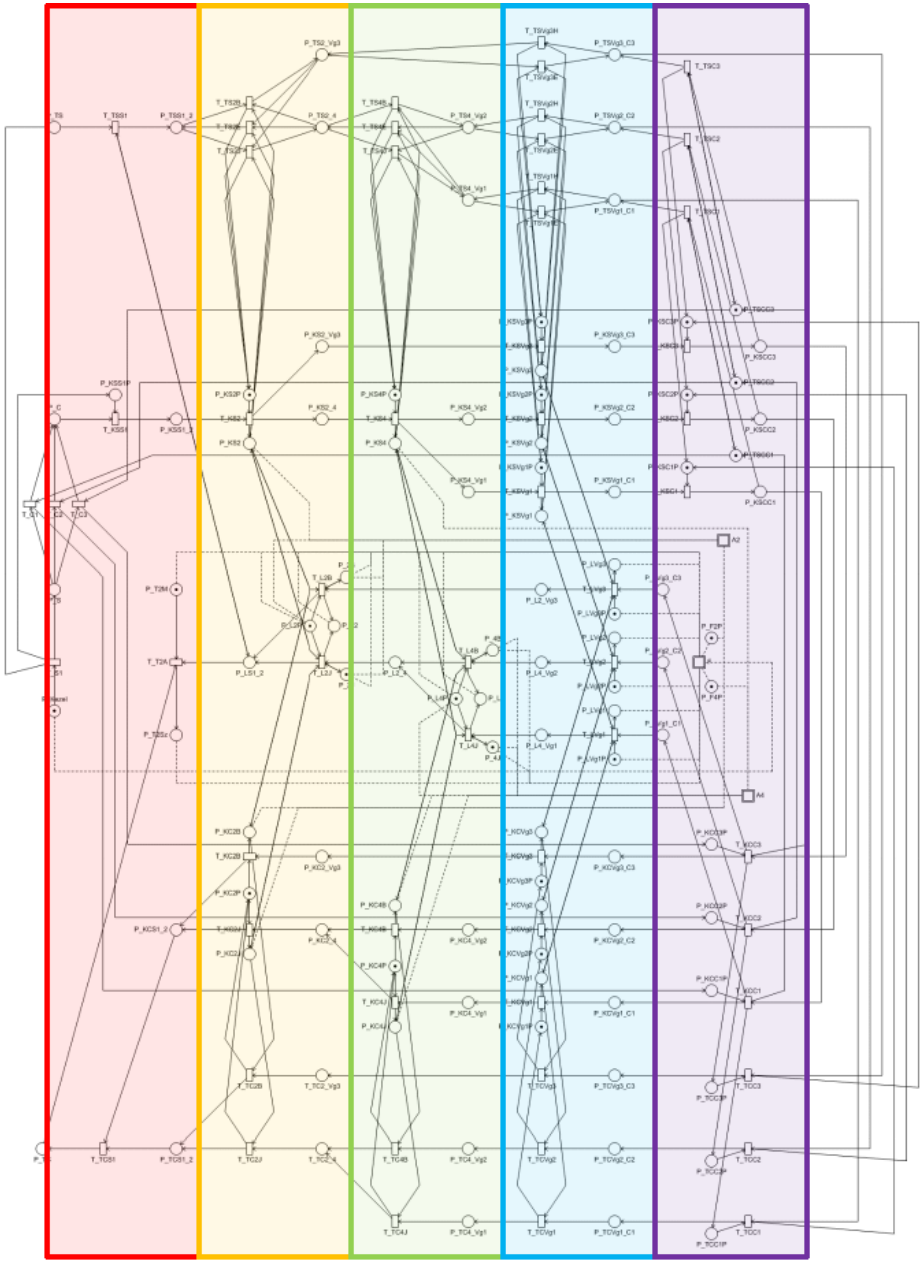


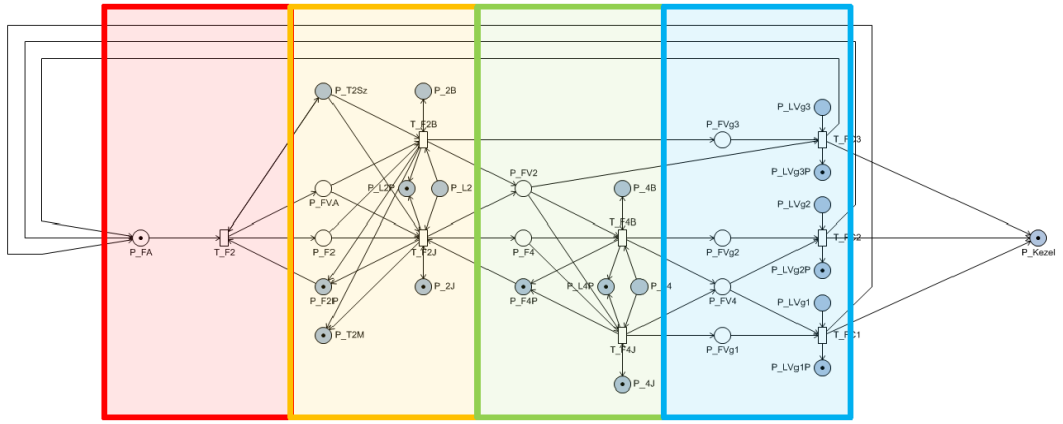
- Three-track junction (two points)
- Shunting signal T2
- Functionality: setting routes (randomly) → train passes → release → repeat
- Some assumptions:
  - Routes only in one direction (Start signal: T2)
  - „Virtual” exit signals
  - No overlap, flank protection needed
  - Conflicting routes are excluded by the nature of the layout
  - Shunting route: no occupation detection needed
  - Only normal operation (no failures)

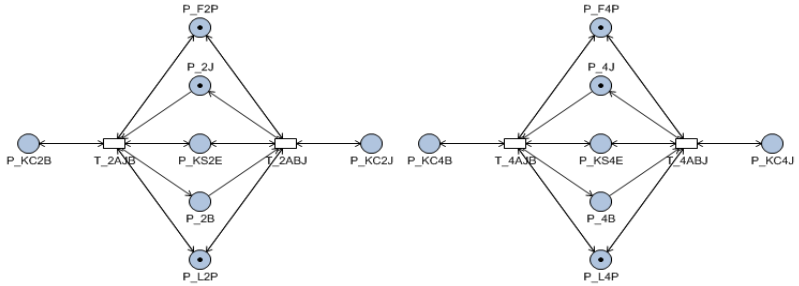



| | A-Tr.1 | A-Tr.2 | A-Tr.3 | Elements | | Signal | Aspect |
|---|---|---|---|---|---|---|---|
| | | | | 2 | 4 | | |
| A-Tr.1 | 1 | | | + | - | T2 | ◎ |
| A-Tr.2 | | 2 | | + | + | T2 | ◎ |
| A-Tr.3 | | | 3 | - | | T2 | ◎ |

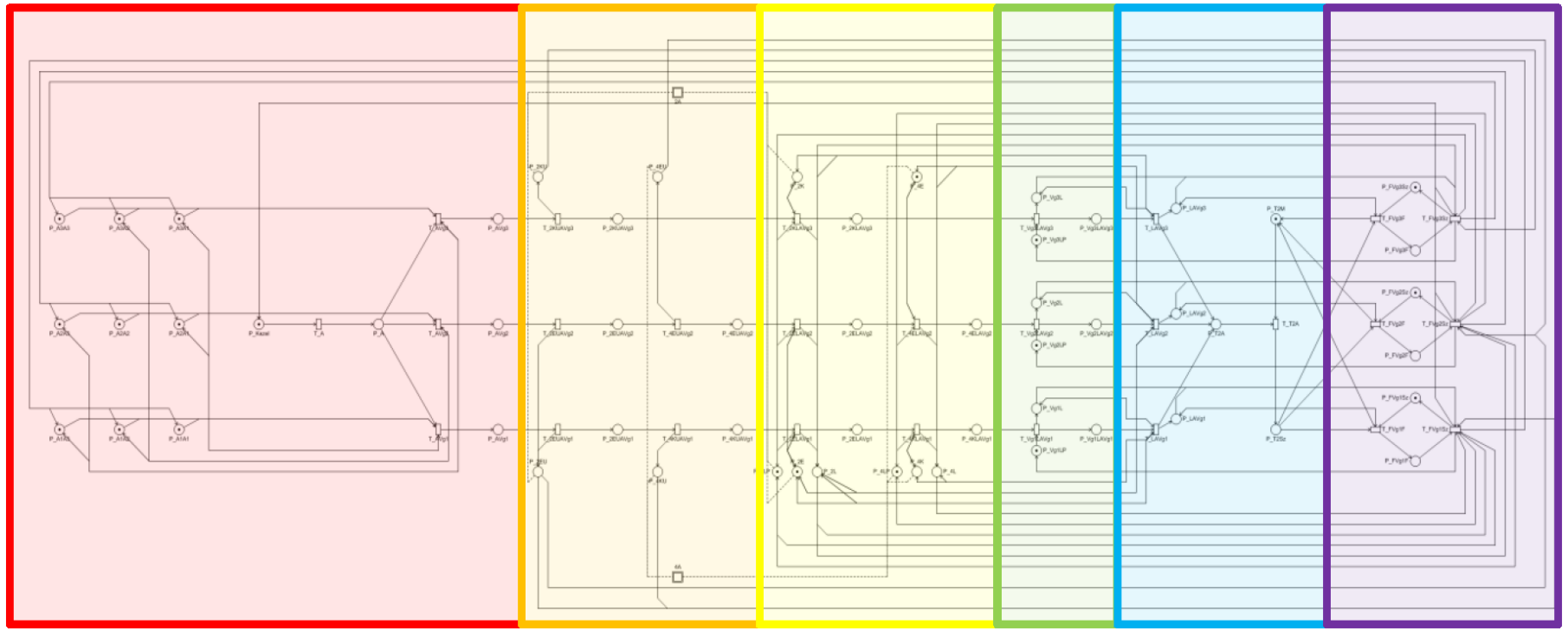
# About the Petri Net Models:
## Geographical Principle

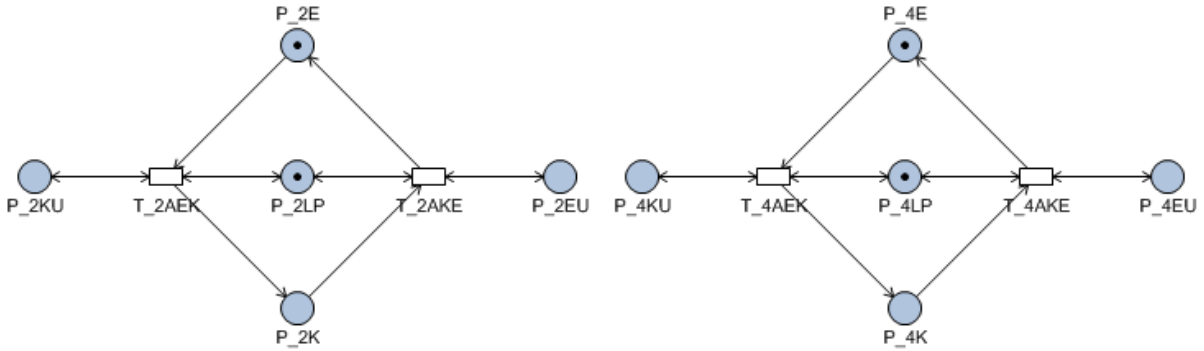




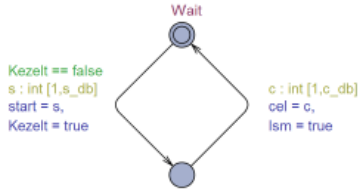



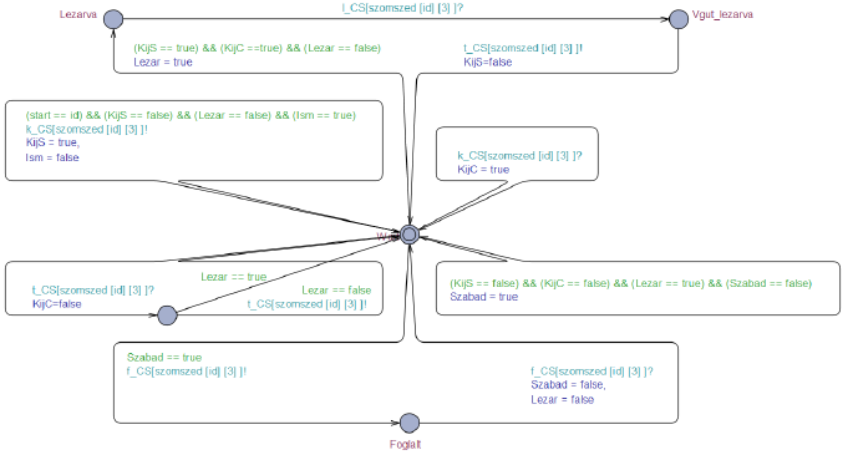
# About the Petri Net Models:
## Tabular Principle

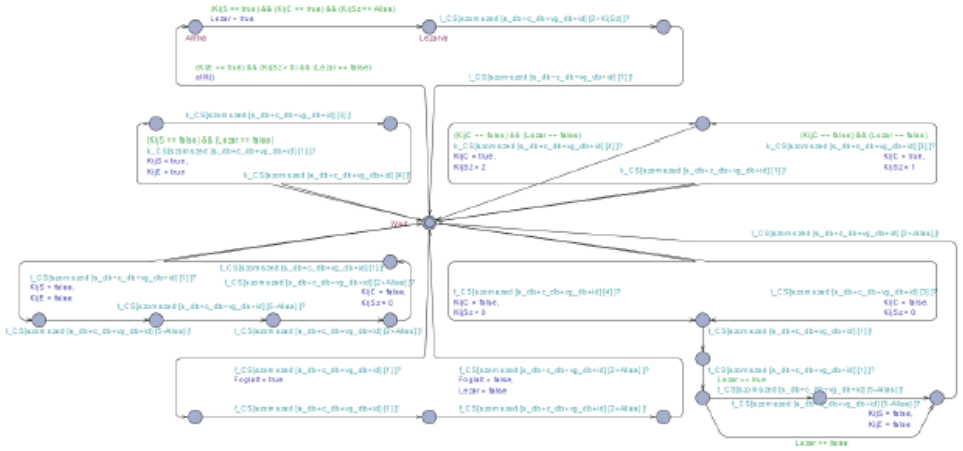


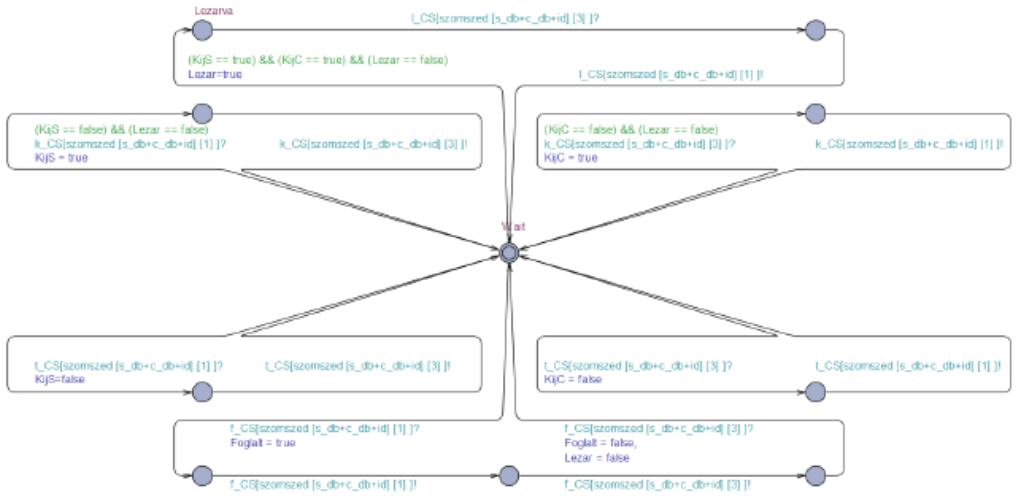


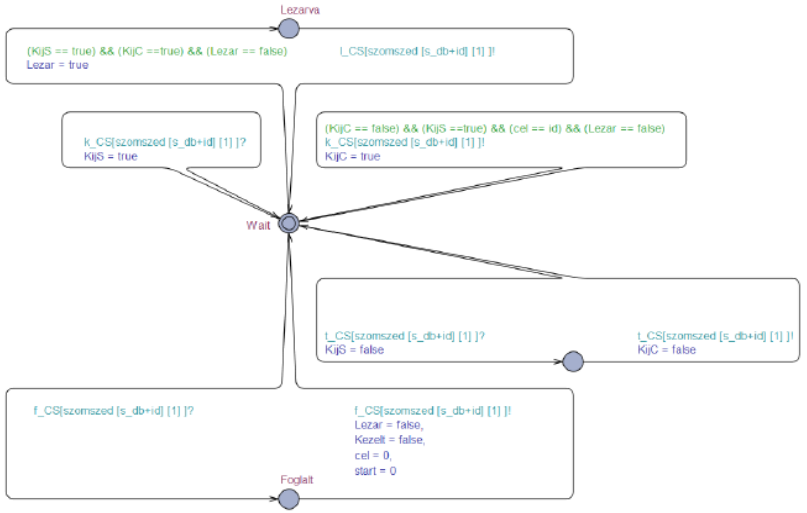
# About the UPPAAL Models:
## Geographical Principle



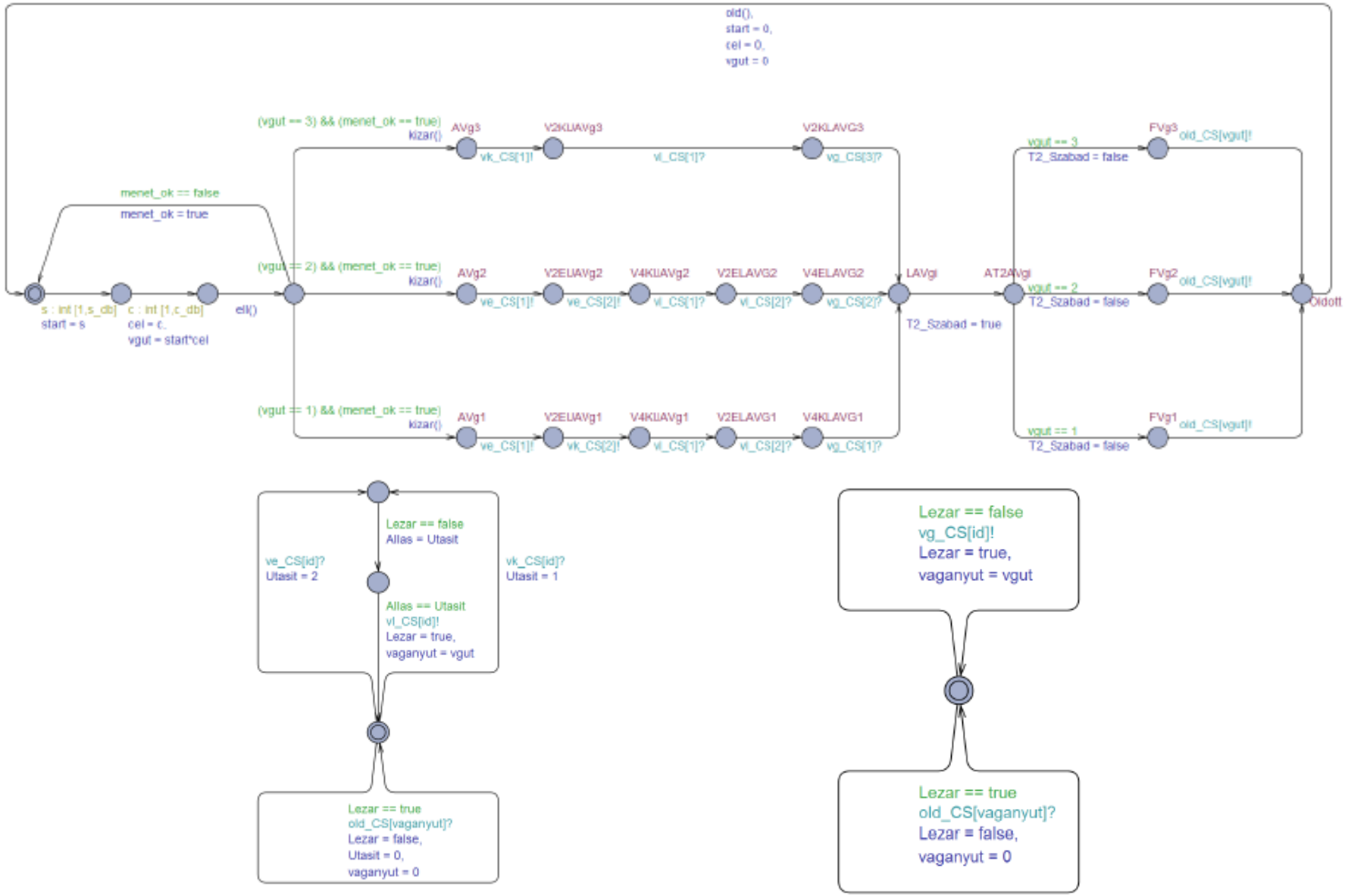





# About the UPPAAL Models:
# Tabular Principle



# Verification – Requirements to check



- Railway interlocking safety property
  - There is interlocking between points and signal:
    - Signal is allowed to set clear only if all points are set and locked
    - Points are allowed to release and set only if the signal is set to stop
- Other functional and structural properties
  - Two-state elements (signal, points) are allowed to be in only one state at a given time
  - No deadlock
  - Boundedness (safety)
  - Exist of home states (no reversibility → points have no initial position)
    - New routes can be set
    - The signal can be set clear
    - The elements are in their „initial” state when choosing the route to be set

# Requirement Refinement



| Nr. | Statement |
|---|---|
| 1. | If the point is occupied, then it is locked. |
| 2. | If the signal shows clear aspect, then the points are set according to the chosen route and they are locked. |
| 3. | A point can be either in normal or reverse position. |
| 4. | A point can be either locked or released. |
| 5. | The signal can show either stop or clear aspect. |
| 6. | If the signal shows clear aspect, then there is a chosen route (or chosen exit). |

| Nr. | Geographical principle, Petri net model |
|---|---|
| 1. | $P\_Fx > 0 \rightarrow P\_Lx > 0$ |
| 2. | $P\_T2Sz > 0 \rightarrow ((P\_xJ/B > 0) \Lambda (P\_Lx > 0))$ |
| 3. | $((P\_xJ > 0) \Lambda (P\_xB = 0)) V ((P\_xJ = 0) \Lambda (P\_xB > 0))$ |
| 4. | $((P\_Lx > 0) \Lambda (P\_LxP = 0)) V ((P\_Lx = 0) \Lambda (P\_LxP > 0))$ |
| 5. | $((P\_T2M > 0) \Lambda (P\_T2Sz = 0)) V ((P\_T2M = 0) \Lambda (P\_T2Sz > 0))$ |
| 6. | $P\_T2Sz > 0 \rightarrow ((P\_LVg1 > 0) V (P\_LVg2 > 0) V (P\_LVg3 > 0))$ |

| Nr. | Geographical principle, UPPAAL model |
|---|---|
| 1. | $Valto(i).Foglalt = true \rightarrow Valto(i).Lezar = true$ |
| 2. | $Start(1).Szabad = true \rightarrow ((Valto(i).Allas = 1/2) \Lambda (Valto(i).Lezar = true))$ |
| 3. | $(Valto(i).Allas = 1) V (Valto(i).Allas = 2)$ |
| 4. | $(Valto(i).Lezar = 0) V (Valto(i).Lezar = 1)$ |
| 5. | $(Start(1).Szabad = 0) V (Start(1).Szabad = 1)$ |
| 6. | $Start(1).Szabad = true \rightarrow 0 < cel \leq 3$ |

| Nr. | Tabular principle, Petri net model |
|---|---|
| 1. | |
| 2. | $P\_T2Sz > 0 \rightarrow ((P\_xE/K > 0) \Lambda (P\_Lx > 0))$ |
| 3. | $((P\_xE > 0) \Lambda (P\_xK = 0)) V ((P\_xE = 0) \Lambda (P\_xK > 0))$ |
| 4. | $((P\_xL > 0) \Lambda (P\_xLP = 0)) V ((P\_xL = 0) \Lambda (P\_xLP > 0))$ |
| 5. | $((P\_T2M > 0) \Lambda (P\_T2Sz = 0)) V ((P\_T2M = 0) \Lambda (P\_T2Sz > 0))$ |
| 6. | $P\_T2Sz > 0 \rightarrow ((P\_LAVg1 > 0) V (P\_LAVg2 > 0) V (P\_LAVg3 > 0))$ |

| Nr. | Tabular principle, UPPAAL model |
|---|---|
| 1. | |
| 2. | $T2\_Szabad = true \rightarrow ((Valto(i).Allas = 1/2) \Lambda (Valto(i).Lezar = true))$ |
| 3. | $(Valto(i).Allas = 1) V (Valto(i).Allas = 2)$ |
| 4. | $(Valto(i).Lezar = 0) V (Valto(i).Lezar = 1)$ |
| 5. | $(T2\_Szabad = 0) V (T2\_Szabad = 1)$ |
| 6. | $T2\_Szabad = true \rightarrow 0 < vgut \leq 3$ |

# Requirements checked - PetriDotNet



| Nr. | Element/ route | CTL expression<br>Geographical principle, Petri net model |
|---|---|---|
| 1. | Point 2 | AG(!(K_L_T.P_F2>0)\|(K_L_T.P_L2>0)) |
| | Point 4 | AG(!(K_L_T.P_F4>0)\|(K_L_T.P_L4>0)) |
| 2. | A-Tr.1 | AG(!((K_L_T.P_T2Sz>0)&(K_L_T.P_LVg1>0))\|((K_L_T.P_2J>0)&(K_L_T.P_L2>0)&(K_L_T.P_4J>0)&(K_L_T.P_L4>0))) |
| | A-Tr.2 | AG(!((K_L_T.P_T2Sz>0)&(K_L_T.P_LVg2>0))\|((K_L_T.P_2J>0)&(K_L_T.P_L2>0)&(K_L_T.P_4B>0)&(K_L_T.P_L4>0))) |
| | A-Tr.3 | AG(!((K_L_T.P_T2Sz>0)&(K_L_T.P_LVg3>0))\|((K_L_T.P_2B>0)&(K_L_T.P_L2>0))) |
| 3. | Point 2 | AG(((K_L_T.P_2B>0)&(K_L_T.P_2J=0))\|((K_L_T.P_2B=0)&(K_L_T.P_2J>0))) |
| | Point 4 | AG(((K_L_T.P_4B>0)&(K_L_T.P_4J=0))\|((K_L_T.P_4B=0)&(K_L_T.P_4J>0))) |
| 4. | Point 2 | AG(((K_L_T.P_L2=0)&(K_L_T.P_L2P>0))\|((K_L_T.P_L2>0)&(K_L_T.P_L2P=0))) |
| | Point 4 | AG(((K_L_T.P_L4=0)&(K_L_T.P_L4P>0))\|((K_L_T.P_L4>0)&(K_L_T.P_L4P=0))) |
| 5. | Signal T2 | AG(((K_L_T.P_T2Sz>0)&(K_L_T.P_T2M=0))\|((K_L_T.P_T2M>0)&(K_L_T.P_T2Sz=0))) |
| 6. | Signal T2 | AG((!(K_L_T.P_T2Sz>0))\|(((K_L_T.P_LVg1>0)&(K_L_T.P_LVg2=0)&(K_L_T.P_LVg3=0))\|((K_L_T.P_LVg1=0)&(K_L_T.P_LVg2>0)&(K_L_T.P_LVg3=0))\|((K_L_T.P_LVg1=0)&(K_L_T.P_LVg2=0)&(K_L_T.P_LVg3>0)))) |

| Nr. | Element/ route | CTL expression<br>Tabular principle, Petri net model |
|---|---|---|
| 1. | Point 2 | |
| | Point 4 | |
| 2. | A-Tr.1 | AG(!((Menet_Elzarasi.P_T2Sz>0)&(Menet_Elzarasi.P_LAVg1>0))\|((Menet_Elzarasi.P_2E>0)&(Menet_Elzarasi.P_2L>0)&(Menet_Elzarasi.P_4K>0)&(Menet_Elzarasi.P_4L>0))) |
| | A-Tr.2 | AG(!((Menet_Elzarasi.P_T2Sz>0)&(Menet_Elzarasi.P_LAVg2>0))\|((Menet_Elzarasi.P_2E>0)&(Menet_Elzarasi.P_2L>0)&(Menet_Elzarasi.P_4E>0)&(Menet_Elzarasi.P_4L>0))) |
| | A-Tr.3 | AG(!((Menet_Elzarasi.P_T2Sz>0)&(Menet_Elzarasi.P_LAVg3>0))\|((Menet_Elzarasi.P_2K>0)&(Menet_Elzarasi.P_2L>0))) |
| 3. | Point 2 | AG(((Menet_Elzarasi.P_2E>0)&(Menet_Elzarasi.P_2K=0))\|((Menet_Elzarasi.P_2E=0)&(Menet_Elzarasi.P_2K>0))) |
| | Point 4 | AG(((Menet_Elzarasi.P_4E>0)&(Menet_Elzarasi.P_4K=0))\|((Menet_Elzarasi.P_4E=0)&(Menet_Elzarasi.P_4K>0))) |
| 4. | Point 2 | AG(((Menet_Elzarasi.P_2L=0)&(Menet_Elzarasi.P_2LP>0))\|((Menet_Elzarasi.P_2L>0)&(Menet_Elzarasi.P_2LP=0))) |
| | Point 4 | AG(((Menet_Elzarasi.P_4L=0)&(Menet_Elzarasi.P_4LP>0))\|((Menet_Elzarasi.P_4L>0)&(Menet_Elzarasi.P_4LP=0))) |
| 5. | Signal T2 | AG(((Menet_Elzarasi.P_T2Sz>0)&(Menet_Elzarasi.P_T2M=0))\|((Menet_Elzarasi.P_T2M>0)&(Menet_Elzarasi.P_T2Sz=0))) |
| 6. | Signal T2 | AG((!(Menet_Elzarasi.P_T2Sz>0))\|(((Menet_Elzarasi.P_LAVg1>0)&(Menet_Elzarasi.P_LAVg2=0)&(Menet_Elzarasi.P_LAVg3=0))\|((Menet_Elzarasi.P_LAVg1=0)&(Menet_Elzarasi.P_LAVg2>0)&(Menet_Elzarasi.P_LAVg3=0))\|((Menet_Elzarasi.P_LAVg1=0)&(Menet_Elzarasi.P_LAVg2=0)&(Menet_Elzarasi.P_LAVg3>0))) |

# Requirements checked - UPPAAL



| Nr. | Element/ route | CTL expression Geographical principle, UPPAAL model |
|---|---|---|
| 1. | Point 2 | A[](Valto(1).Foglalt!=true)\|\|(Valto(1).Lezar==true)) |
| | Point 4 | A[](Valto(2).Foglalt!=true)\|\|(Valto(2).Lezar==true)) |
| 2. | A-Tr.1 | A[](!((Start(1).Szabad==1)&&(cel==1))\|\|((Valto(1).Allas==2)&& (Valto(1).Lezar==1)&&(Valto(2).Allas==2)&&(Valto(2).Lezar==1))) |
| | A-Tr.2 | A[](!((Start(1).Szabad==1)&&(cel==2))\|\|((Valto(1).Allas==2)&& (Valto(1).Lezar==1)&&(Valto(2).Allas==1)&&(Valto(2).Lezar==1))) |
| | A-Tr.3 | A[](!((Start(1).Szabad==1)&&(cel==3))\|\|((Valto(1).Allas==1)&& (Valto(1).Lezar==1))) |
| 3. | Point 2 | A[](Valto(1).Allas==1)\|\|(Valto(1).Allas==2)) |
| | Point 4 | A[](Valto(2).Allas==1)\|\|(Valto(2).Allas==2)) |
| 4. | Point 2 | A[](Valto(1).Lezar==0)\|\|(Valto(1).Lezar==1)) |
| | Point 4 | A[](Valto(2).Lezar==0)\|\|(Valto(2).Lezar==1)) |
| 5. | Signal T2 | A[](Start(1).Szabad==0)\|\|(Start(1).Szabad==1)) |
| 6. | Signal T2 | A[]((!(Start(1).Szabad==1))\|\|(cel>>0))) |

| Nr. | Element/ route | CTL expression Tabular principle, UPPAAL model |
|---|---|---|
| 1. | Point 2 | |
| | Point 4 | |
| 2. | A-Tr.1 | A[](!((Menet_Elzarasi.T2_Szabad==1)&&(cel==1))\|\|((Valto(1).Allas==2)&& (Valto(1).Lezar==1)&&(Valto(2).Allas==1)&&(Valto(2).Lezar==1))) |
| | A-Tr.2 | A[](!((Menet_Elzarasi.T2_Szabad==1)&&(cel==2))\|\|((Valto(1).Allas==2)&& (Valto(1).Lezar==1)&&(Valto(2).Allas==2)&&(Valto(2).Lezar==1))) |
| | A-Tr.3 | A[](!((Menet_Elzarasi.T2_Szabad==1)&&(cel==3))\|\|((Valto(1).Allas==1)&& (Valto(1).Lezar==1))) |
| 3. | Point 2 | A[](Valto(1).Allas==1)\|\|(Valto(1).Allas==2)) |
| | Point 4 | A[](Valto(2).Allas==1)\|\|(Valto(2).Allas==2)) |
| 4. | Point 2 | A[](Valto(1).Lezar==0)\|\|(Valto(1).Lezar==1)) |
| | Point 4 | A[](Valto(2).Lezar==0)\|\|(Valto(2).Lezar==1)) |
| 5. | Signal T2 | A[](Menet_Elzarasi.T2_Szabad==0)\|\|(Menet_Elzarasi.T2_Szabad==1)) |
| 6. | Signal T2 | A[]((!(Menet_Elzarasi.T2_Szabad==1))\|\|(cel>>0))) |

# Requirements checked – Other Properties



| Property | Petri net | |
|---|---|---|
| | geographical | tabular |
| Safety | AG(K_L_T.P_*x*<=1)<br>AG(A2.P_*x*<=1)<br>AG(A4.P_*x*<=1)<br>AG(F.P_*x*<=1) | AG(Menet_Elzarasi.P_x<=1)<br>AG(A2.P_x<=1)<br>AG(A4.P_x<=1) |
| New routes can be set | AG(EF(K_L_T.P_Kezel>0)) | AG(EF(Menet_Elzarasi.P_Kezel>0)) |
| The signal can be set clear | AG(EF(K_L_T.P_T2Sz>0)) | AG(EF(Menet_Elzarasi.P_T2Sz>0)) |
| The elements are in their initial state when choosing the route to be set | AG((!(K_L_T.P_Kezel>0))\|<br>((K_L_T.P_L2=0)&<br>(K_L_T.P_L4=0)&<br>(K_L_T.P_LVg1=0)&<br>(K_L_T.P_LVg2=0)&<br>(K_L_T.P_LVg3=0)&<br>(K_L_T.P_KS2E=0)&<br>(K_L_T.P_KS4E=0)&<br>(K_L_T.P_KSVg1=0)&<br>(K_L_T.P_KSVg2=0)&<br>(K_L_T.P_KSVg3=0)&<br>(K_L_T.P_KC2J=0)&<br>(K_L_T.P_KC2B=0)&<br>(K_L_T.P_KC4J=0)&<br>(K_L_T.P_KC4B=0)&<br>(K_L_T.P_KCVg1=0)&<br>(K_L_T.P_KCVg2=0)&<br>(K_L_T.P_KCVg3=0)&<br>(K_L_T.P_T2Sz=0))) | AG((!(Menet_Elzarasi.P_Kezel>0))\|<br>((Menet_Elzarasi.P_2L=0)&<br>(Menet_Elzarasi.P_4L=0)&<br>(Menet_Elzarasi.P_Vg1L=0)&<br>(Menet_Elzarasi.P_Vg2L=0)&<br>(Menet_Elzarasi.P_Vg3L=0)&<br>(Menet_Elzarasi.P_2EU=0)&<br>(Menet_Elzarasi.P_2KU=0)&<br>(Menet_Elzarasi.P_4EU=0)&<br>(Menet_Elzarasi.P_4KU=0)&<br>(Menet_Elzarasi.P_LAVg1=0)&<br>(Menet_Elzarasi.P_LAVg2=0)&<br>(Menet_Elzarasi.P_LAVg3=0)&<br>(Menet_Elzarasi.P_T2Sz=0))) |

| Property | UPPAAL | |
|---|---|---|
| | geographical | tabular |
| Deadlock freedom | A[] not deadlock | A[] not deadlock |
| The elements are in their initial state when choosing the route to be set | A[](!((start==0)&&(cel==0)))\|\|<br>((Start(1).Lezar==0)&&<br>(Valto(1).Lezar==0)&&<br>(Valto(2).Lezar==0)&&<br>(Vagany(1).Lezar==0)&&<br>(Vagany(2).Lezar==0)&&<br>(Vagany(3).Lezar==0)&&<br>(Cel(1).Lezar==0)&&<br>(Cel(2).Lezar==0)&&<br>(Cel(3).Lezar==0)&&<br>(Start(1).KijS==0)&&<br>(Valto(1).KijS==0)&&<br>(Valto(2).KijS==0)&&<br>(Vagany(1).KijS==0)&&<br>(Vagany(2).KijS==0)&&<br>(Vagany(3).KijS==0)&&<br>(Cel(1).KijS==0)&&<br>(Cel(2).KijS==0)&&<br>(Cel(3).KijS==0)&&<br>(Start(1).KijC==0)&&<br>(Valto(1).KijC==0)&&<br>(Valto(2).KijC==0)&&<br>(Vagany(1).KijC==0)&&<br>(Vagany(2).KijC==0)&&<br>(Vagany(3).KijC==0)&&<br>(Cel(1).KijC==0)&&<br>(Cel(2).KijC==0)&&<br>(Cel(3).KijC==0)&&<br>(Valto(1).KijE==0)&&<br>(Valto(2).KijE==0)&&<br>(Valto(1).KijSz==0)&&<br>(Valto(2).KijSz==0)&&<br>(Start(1).Szabad==0))) | A[](!((start==0)&&(cel==0)))\|\|<br>((Valto(1).Lezar==0)&&<br>(Valto(2).Lezar==0)&&<br>(Vagany(1).Lezar==0)&&<br>(Vagany(2).Lezar==0)&&<br>(Vagany(3).Lezar==0)&&<br>(Valto(1).Utasit==0)&&<br>(Valto(2).Utasit==0)&&<br>(Menet_Elzarasi.T2_Szabad==0))) |

# Verification Results



| Nr. | Element/ route | Petri net | | UPPAL | |
|---|---|---|---|---|---|
| | | geographical | tabular | geographical | tabular |
| 1. | Point 2 | true | | true | |
| | Point 4 | true | | true | |
| 2. | A-Tr.1 | true | true | true | true |
| | A-Tr.2 | true | true | true | true |
| | A-Tr.3 | true | true | true | true |
| 3. | Point 2 | true | true | true | true |
| | Point 4 | true | true | true | true |
| 4. | Point 2 | true | true | true | true |
| | Point 4 | true | true | true | true |
| 5. | Signal T2 | true | true | true | true |
| 6. | Signal T2 | true | true | true | true |

| Property | Petri net | | UPPAAL | |
|---|---|---|---|---|
| | geographical | tabular | geographical | tabular |
| Deadlock freedom | | | true | true |
| Safety | true | true | | |
| New routes can be set | true | true | | |
| The signal can be set clear | true | true | | |
| The elements are in their initial state when choosing the route to be set | true | true | true | true |

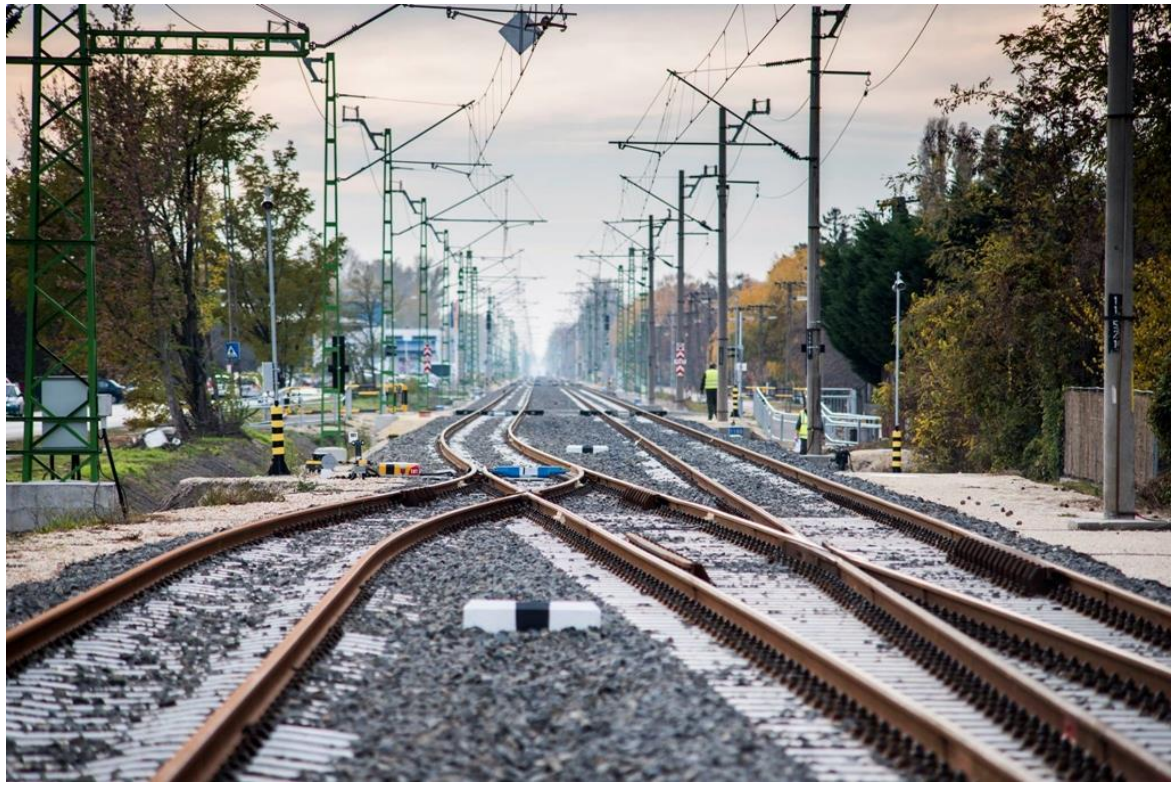
| Comparing viewpoint | PetriDotNet | UPPAAL |
|---|---|---|
| GUI | ✔ | ✔ |
| Easy positioning of building elements | ✔ | ✔ |
| Manual simulation | ✔ | ✔ |
| Automatic simulation | ✔ | ✔ |
| Help | X | ✔ |
| Built in model checker | ✔ | ✔ |
| Interoperability with other tools | ✔ | X |
| Export model as image | ✔ | X |

| Comparing viewpoint | | PetriDotNet | UPPAAL |
|---|---|---|---|
| Applicable operators | AG, AF | ✔ | ✔ |
| | EG, EF | ✔ | ✔ |
| | AU, EU | ✔ | X |
| | AX, EX | ✔ | X |
| | --> („leads to") | X | ✔ |
| General property analysis of the model | | ✔ | X |
| CTL expression for deadlock freedom | | X | ✔ |
| Expression editor | | ✔ | X |
| Saving expressions with the model | | X | ✔ |
| Feedback about the result of checking | | ✔ | ✔ |
| Counterexample generation | | X | ✔ |

# Actual Research



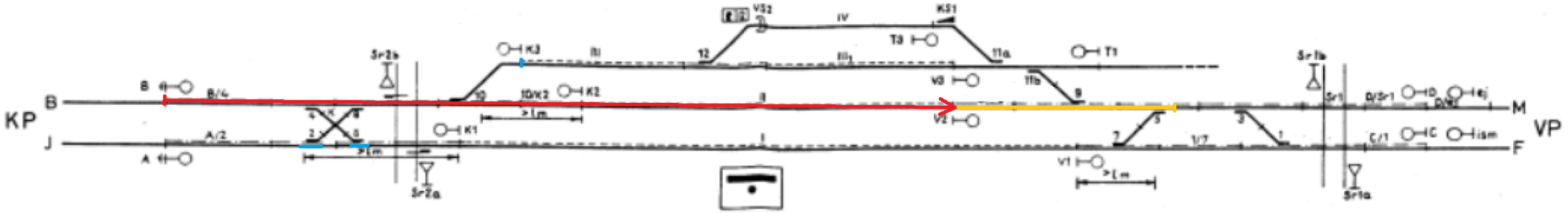
- Providing railway interlocking plans' correctness
- All interlocking components know how to behave correctly in a given „situation" → we tell them the situation
- Finite number of elements + finite number of an element's state → finite state space



# Planning Tasks

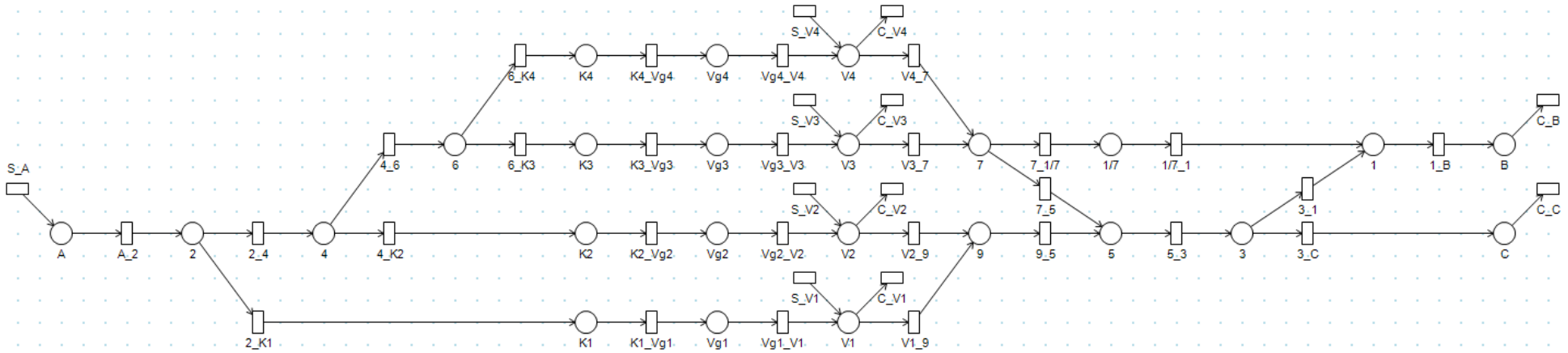


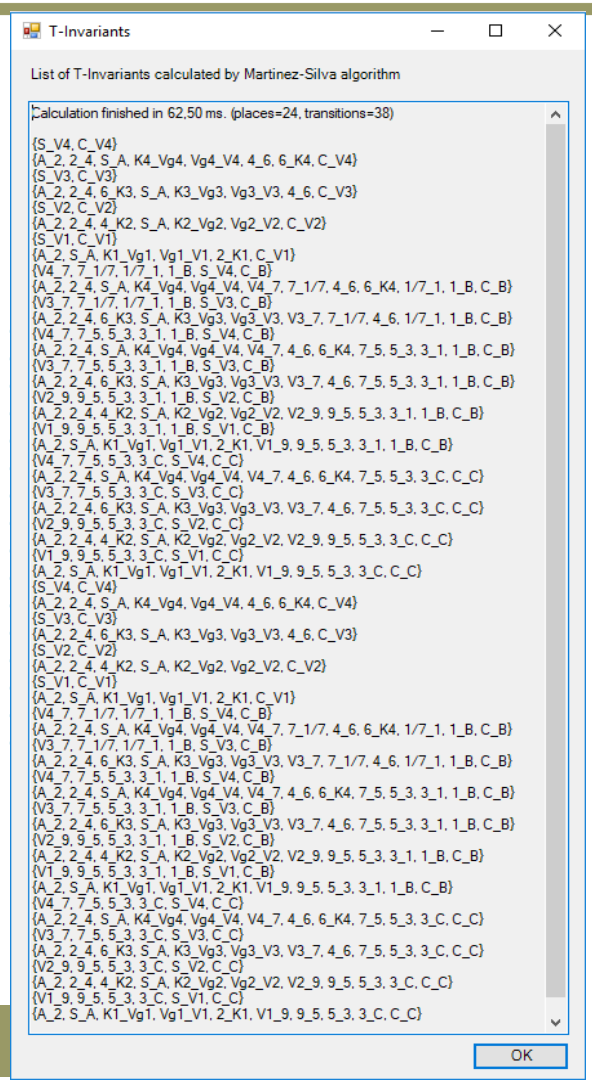
- Application of the general interlocking
- Finding possible routes from one signal to the other
- Choosing the optimal route (if there are more possible between two signals) → not safety-critical
- Determining the possible aspects of the start signal
- Flank protection
- Overlap



# An example – use of T-invariants





# Thank you for your attention!