Heinrich Heine
Universität
Düsseldorf

Gefördert durch:

Bundesministerium
für Wirtschaft
und Energie

aufgrund eines Beschlusses
des Deutschen Bundestages

A Formal Model of Train Control with AI-based Obstacle Detection

RSSR’2023, Berlin

Jan Gruteser, David Geleßus, Michael Leuschel, **Jan Roßbach**, **Fabian Vu**

12.10.2023

Motivation

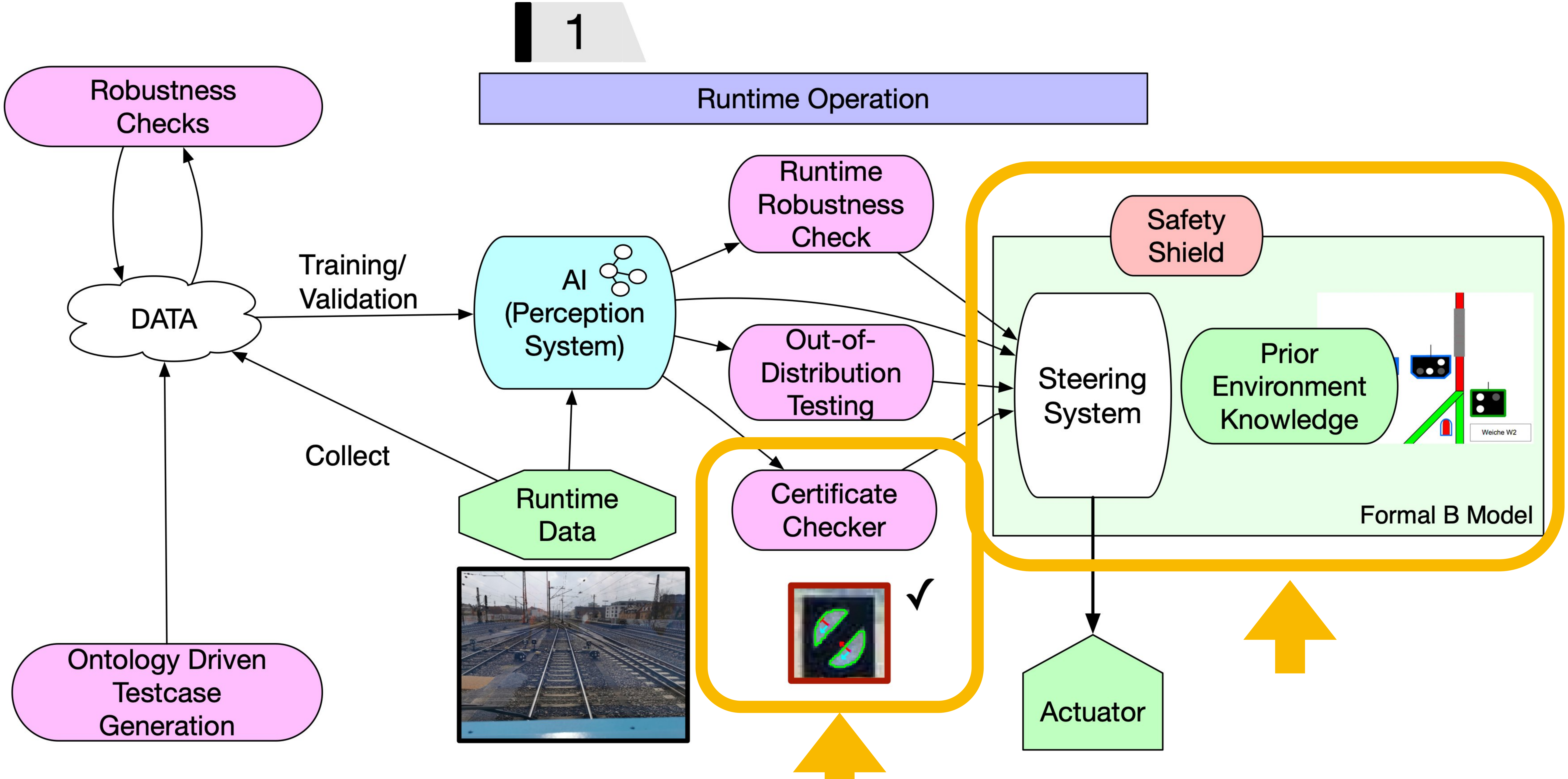
- AI becomes more and more important - e.g. autonomous railway systems

- Hard to verify with classical approaches like formal methods

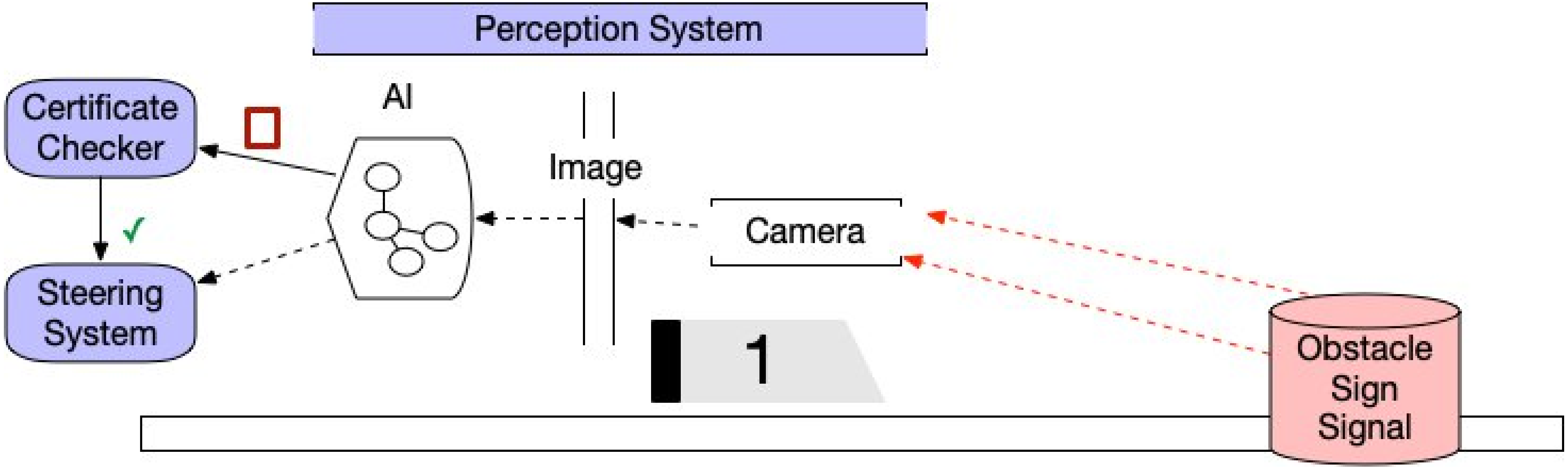
- Goals/Challenges:
  - Verification & Certification of autonomous railway systems
  - Study impact of AI failures
  - Establish effective counter-measures

Measures towards certification

for AI-based obstacle detection

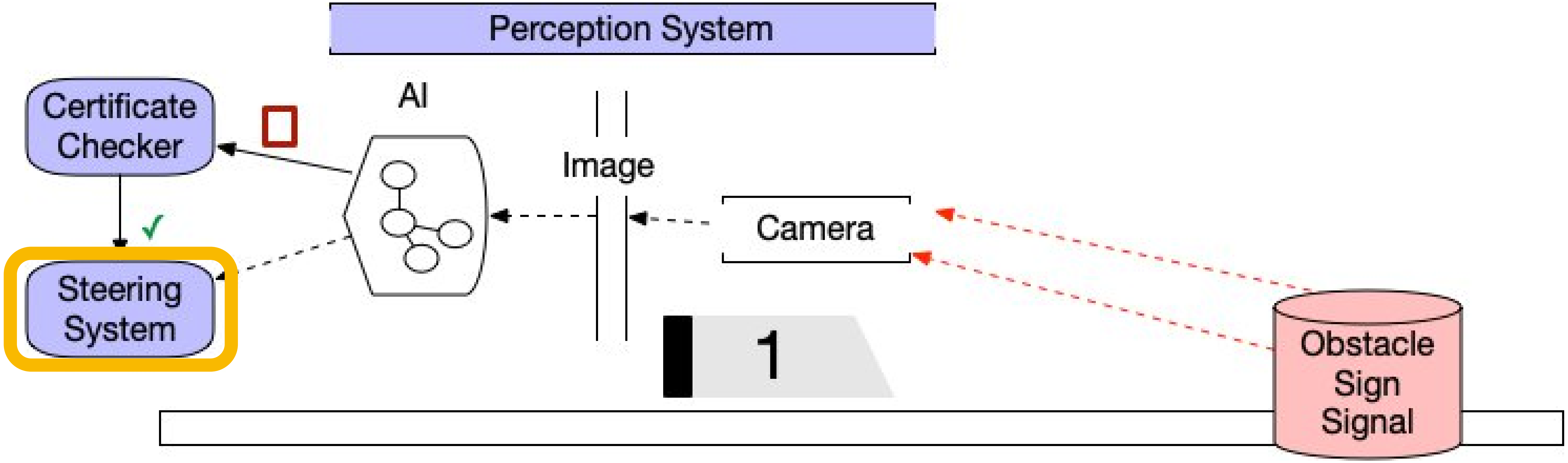


Case Study

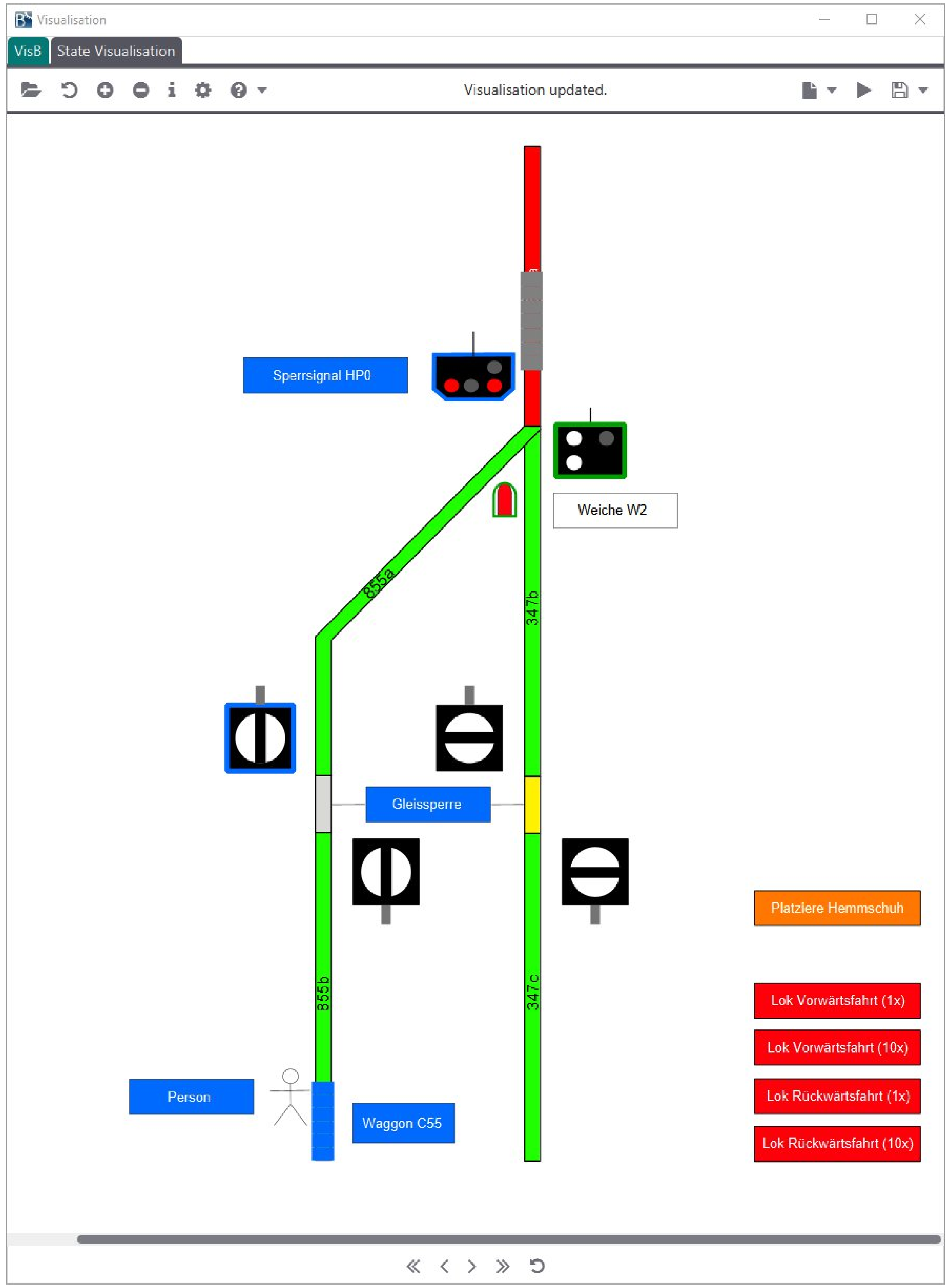

Perception System
Certificate Checker
□
✓
Steering System
AI
Image
Camera
1
Obstacle Sign Signal

Idea 1: Formal B Model

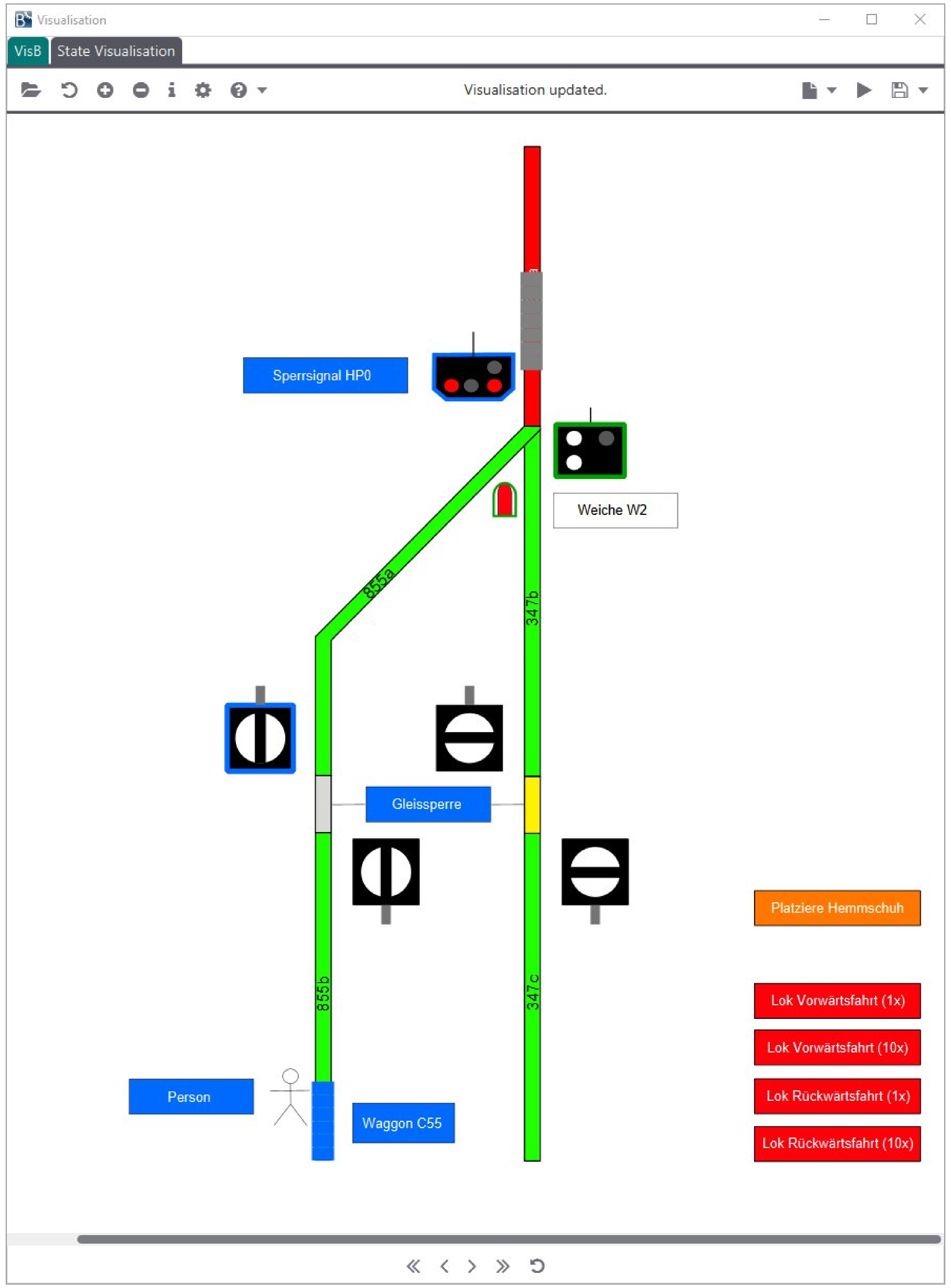
Study impact of AI for behavior of overall system



Formal Model: DEMO



Formal Model


Visualisation updated.

Requirements (abbreviated):

- **Mission Order**: Drive along route - recognize all objects correctly
- **REC1-5**: Perception system must recognize all objects better than humans
- **SAF1-5**: When all objects are recognized correctly - there are no accidents
- **PROP1**: Safety-critical situations shall occur less frequently than with humans
- **PROP2**: Probability of achieving Mission Order shall better than with humans

Validation and Verification

- **Mission Order**
  - validated by traces with different variations for correct/incorrect/non-detection
  - evaluate impact of AI flaws

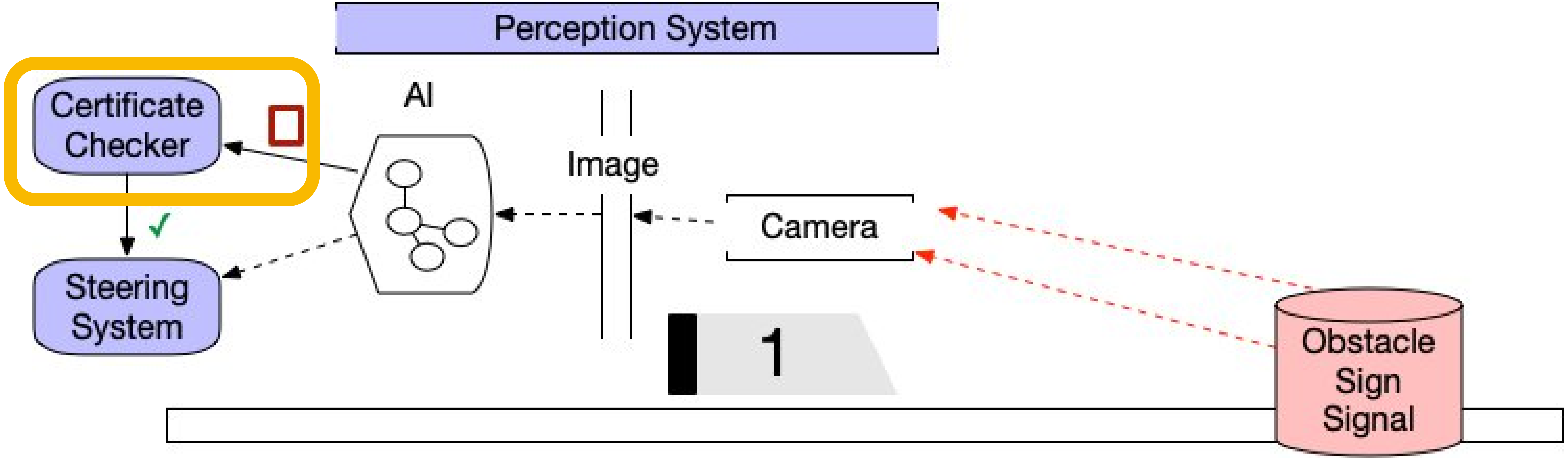
- **SAF1-5** - validated by LTL model checking - on reduced models

- **PROP1 and PROP2**
  - validated by simulation + hypothesis testing (artificial values for probabilities)
  - estimate likelihood of accidents

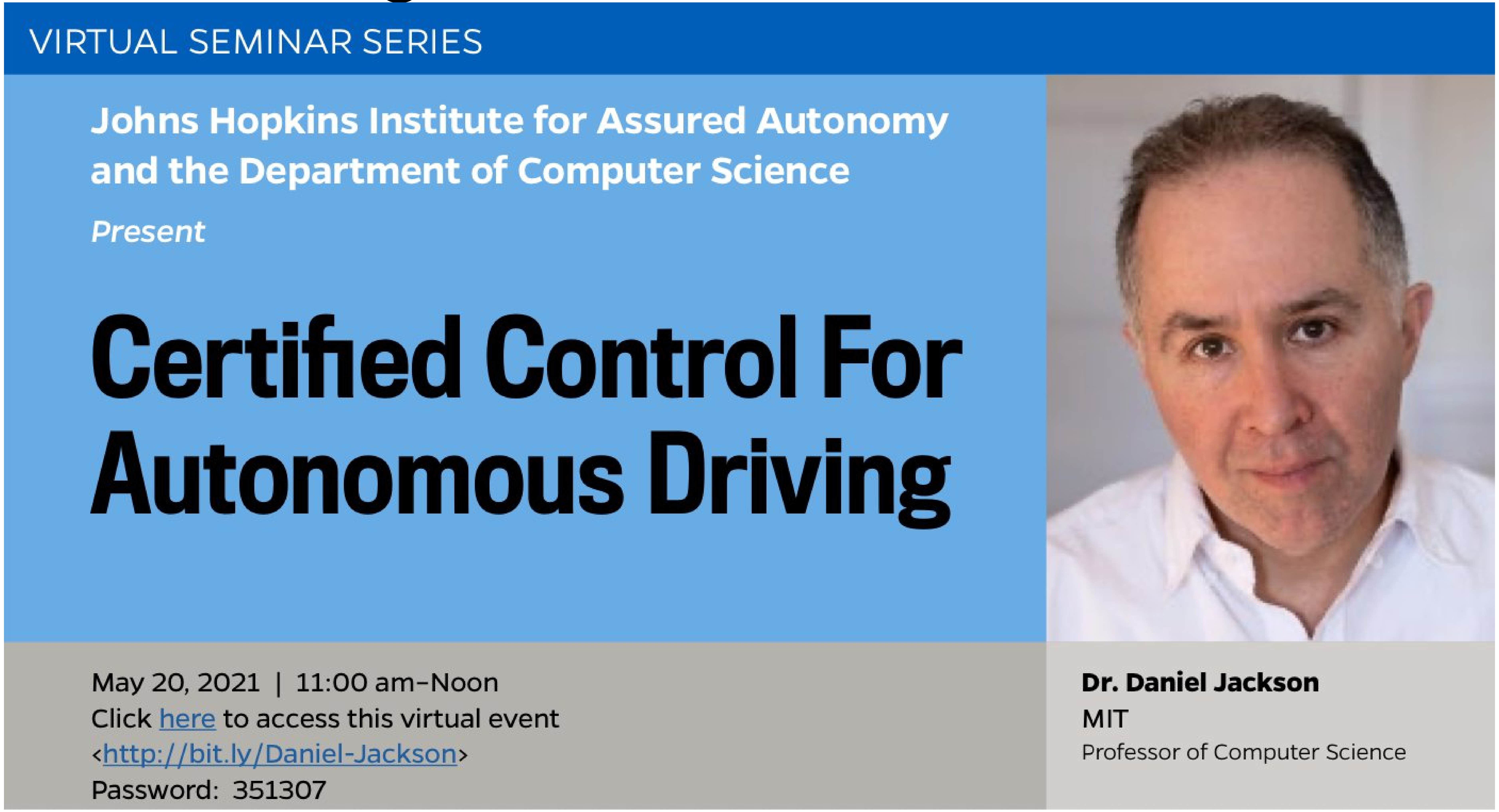
- **REC1-5 - certificate checking**; hard to verify with formal methods

Idea 2: Certified Control

Runtime Monitoring/Verification of AI Object Detection



Origin of “Certified Control”



VIRTUAL SEMINAR SERIES

Johns Hopkins Institute for Assured Autonomy and the Department of Computer Science

Present

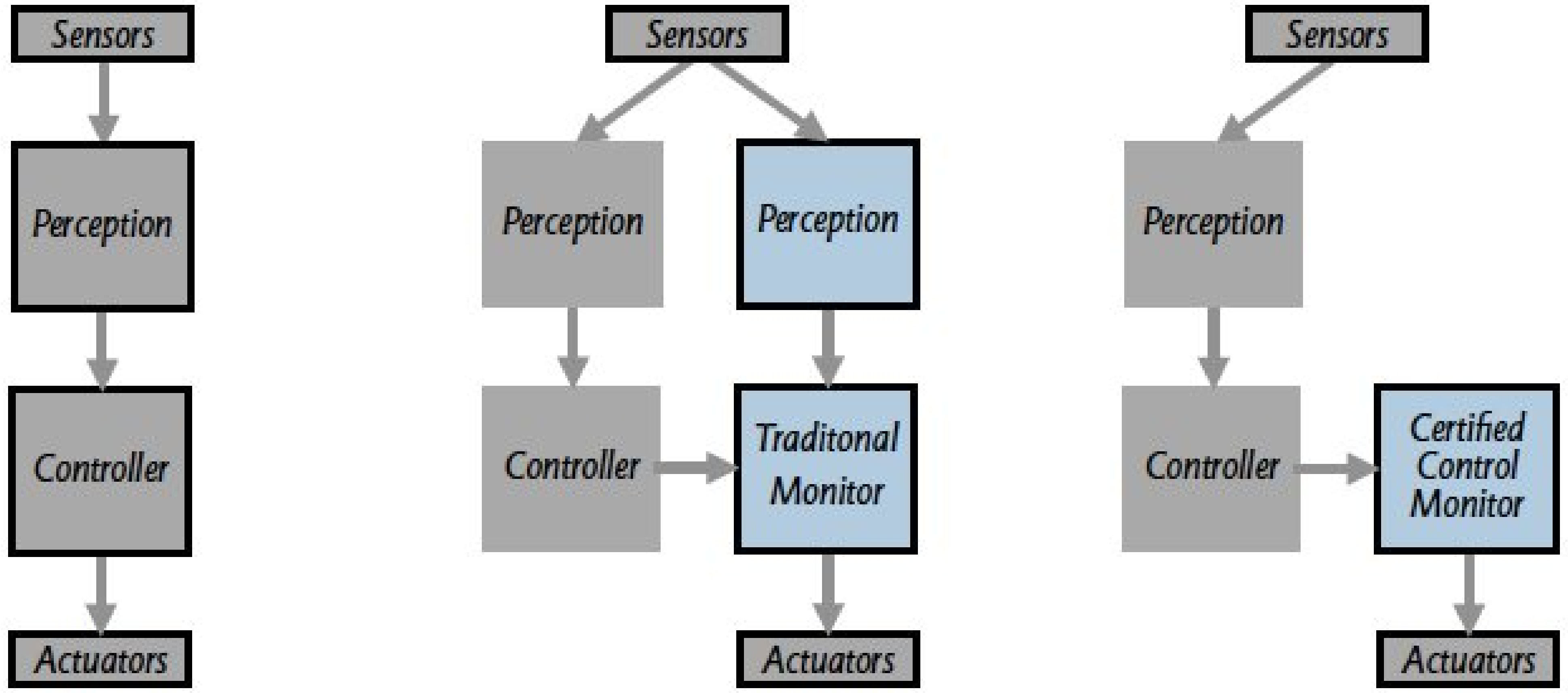
Certified Control For Autonomous Driving

May 20, 2021 | 11:00 am–Noon
Click here to access this virtual event
<http://bit.ly/Daniel-Jackson>
Password: 351307

Dr. Daniel Jackson
MIT
Professor of Computer Science

https://iaa.jhu.edu/event/iaa-seminar-series-daniel-jackson-mit/

Certified Control



- Main perception and control subsystems provide a **certificate**
- Certificate is checked by a verified certificate checker (trusted base)
- Allows for verification of perception system properties, without AI verification
- Do we need an entirely new kind of AI? Not always!
- We developed a prototype for one specific use case using only existing Models

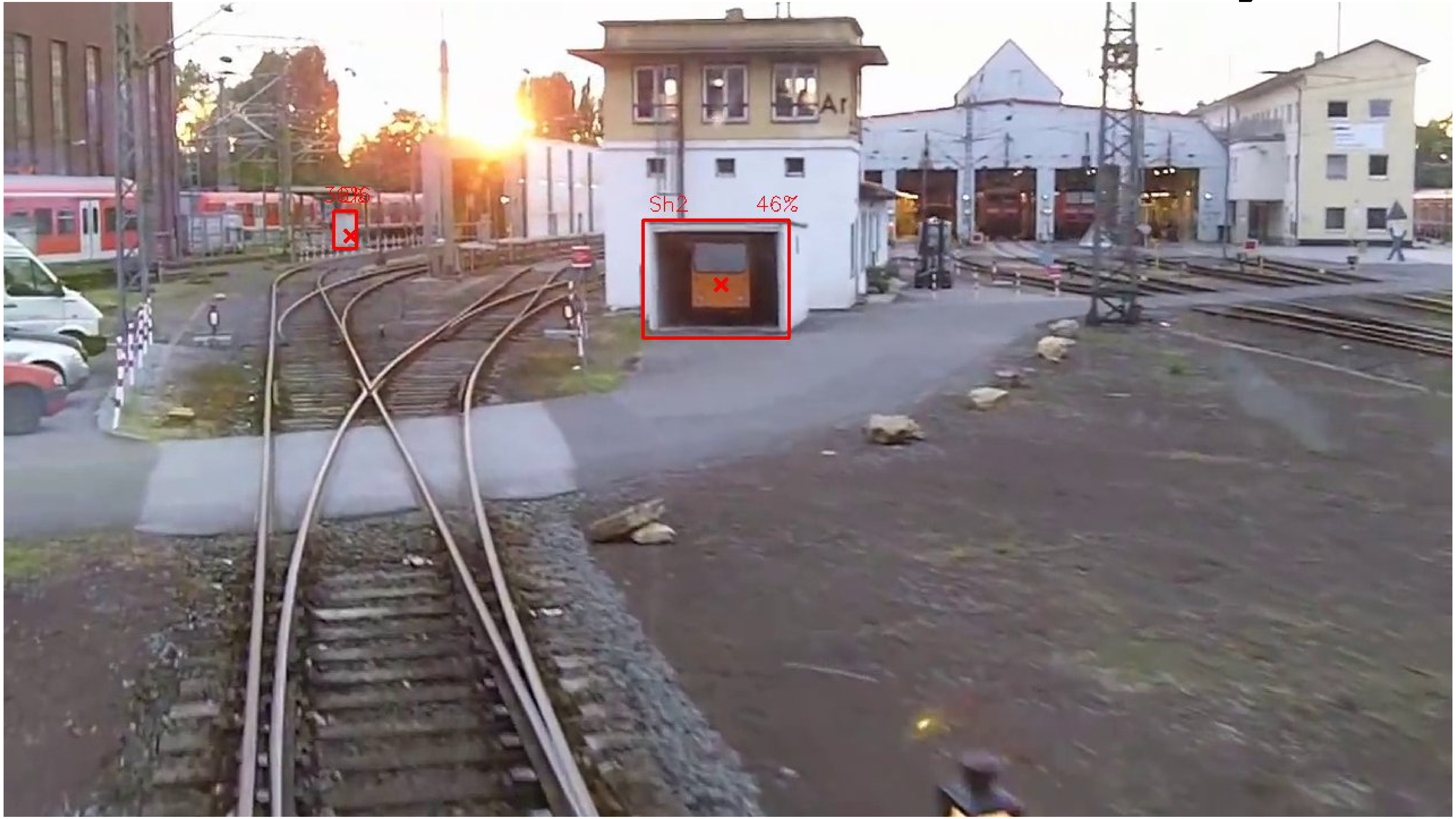
Idea 2: Certified Control

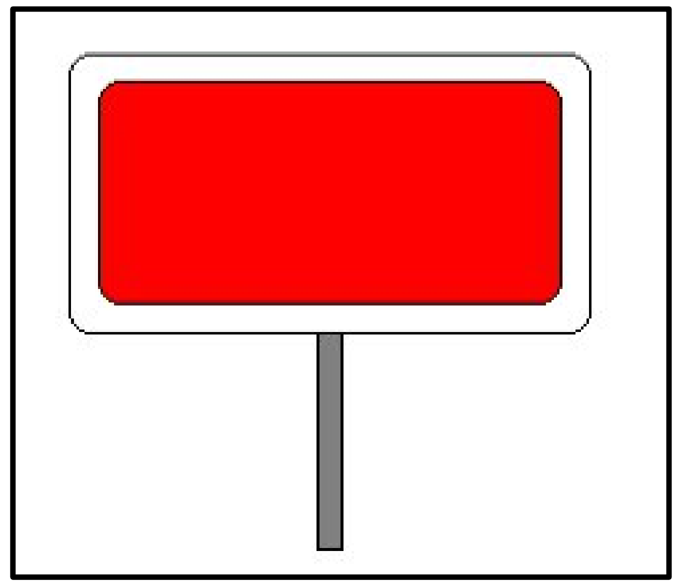
Implement and evaluate it in the context of our case study

Sh1
Bild
KI
Output
Certificate
Kamera
1
YOLO v8 Model trained for shunting signs
Hindernis
Certificate Checker
OK
Steering System
OpenCV based feature detection
Sh1

Certificate = formally verifiable explanation

Erroneous classification detected by checker:

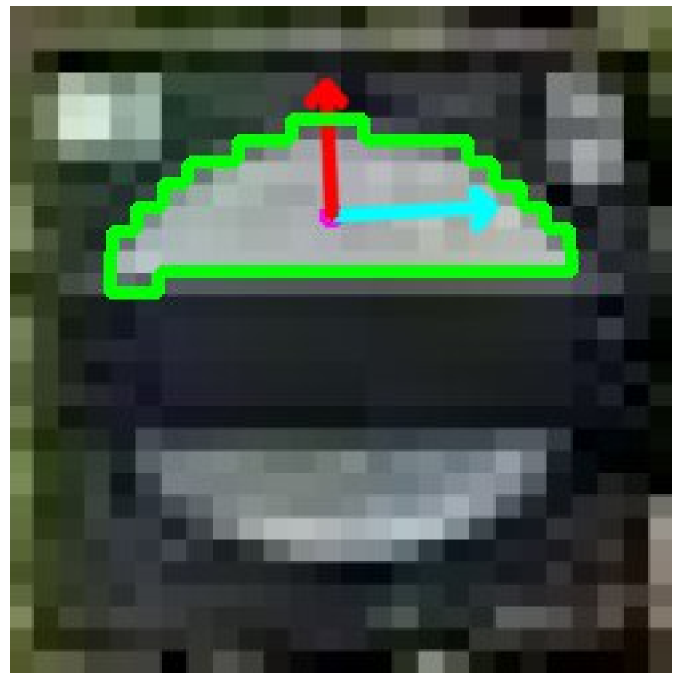


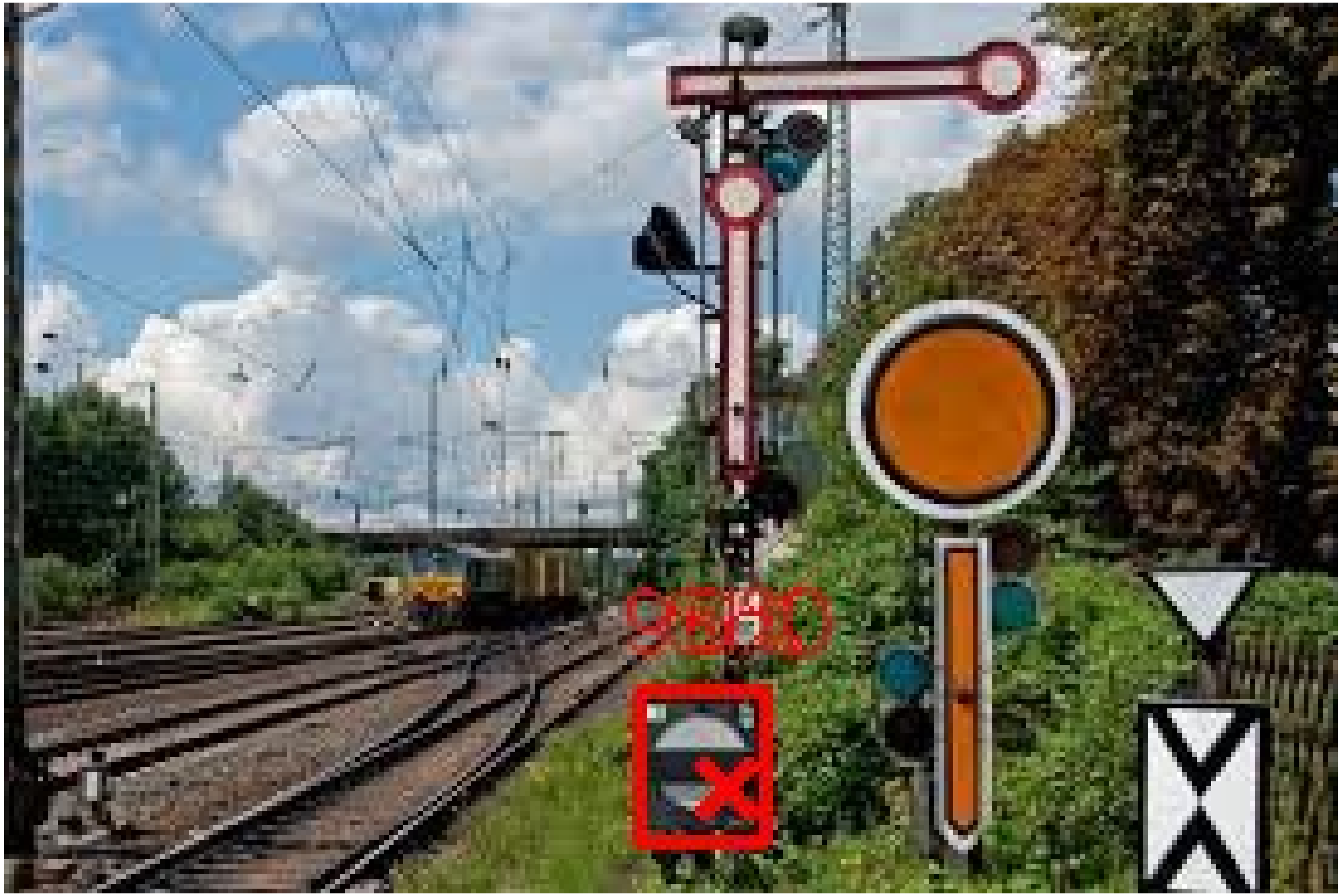


Signal Sh2
Protective STOP

Correct classification rejected by checker:

the certificate
checker may
also reject
correct classifications





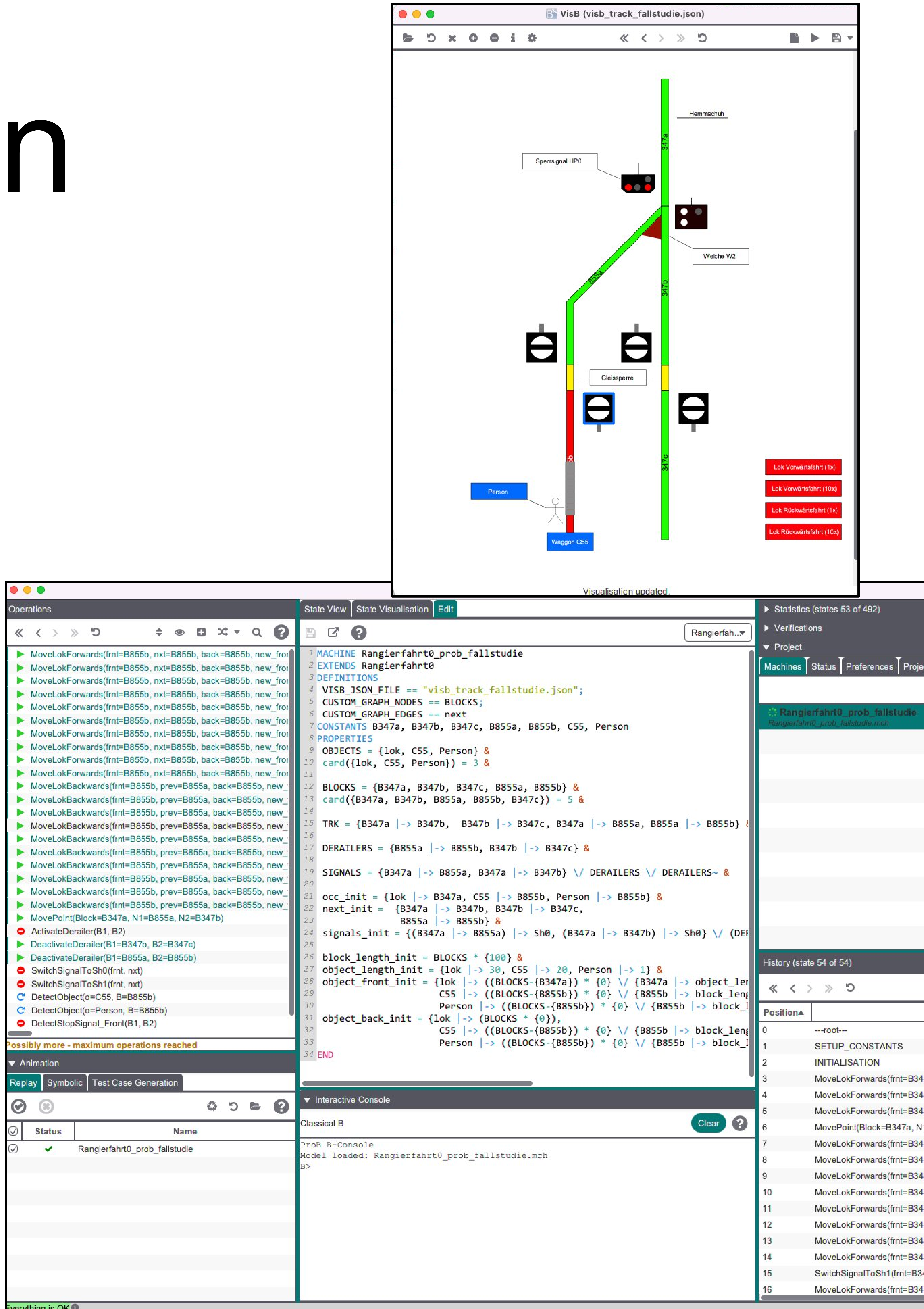
Certified Control: Challenges/Problems

- Not a proof of correct classification results
- Gain Accuracy at the cost of Recall (Trade off depends on the exact implementation)
- Individual Solution for each safety property required => expensive
- Does not address non detections (false negatives)

Solutions - if sign is not detected by AI:
1. provide location of signals/signs - go into a safe mode - when no sign detected where expected,
2. or accept AI can make errors - conduct a probabilistic analysis (SimB)

Conclusion

- Tooling with **ProB**
- Validation:
  - Check individual scenarios - evaluate impact of AI flaws
  - Run simulations with various assumptions - estimate likelihood of accidents
- Verification:
  - Model checking – on reduced models
  - Formal proof still challenging
  - Certificate checking for perception system



Visualisation updated.

```
MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_fro
MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_fro
MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_fro
MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_fro
MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_fro
MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_fro
MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_fro
MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_fro
MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_fro
MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_fro
MoveLokBackwards(frnt=B855b, prev=B855a, back=B855b, new_
MoveLokBackwards(frnt=B855b, prev=B855a, back=B855b, new_
MoveLokBackwards(frnt=B855b, prev=B855a, back=B855b, new_
MoveLokBackwards(frnt=B855b, prev=B855a, back=B855b, new_
MoveLokBackwards(frnt=B855b, prev=B855a, back=B855b, new_
MoveLokBackwards(frnt=B855b, prev=B855a, back=B855b, new_
MoveLokBackwards(frnt=B855b, prev=B855a, back=B855b, new_
MoveLokBackwards(frnt=B855b, prev=B855a, back=B855b, new_
MoveLokBackwards(frnt=B855b, prev=B855a, back=B855b, new_
MoveLokBackwards(frnt=B855b, prev=B855a, back=B855b, new_
MovePoint(Block=B347a, N1=B855a, N2=B347b)
ActivateDerailer(B1, B2)
DeactivateDerailer(B1=B347b, B2=B347c)
DeactivateDerailer(B1=B855a, B2=B855b)
SwitchSignalToSh0(frnt, nxt)
SwitchSignalToSh1(frnt, nxt)
DetectObject(o=C55, B=B855b)
DetectObject(o=Person, B=B855b)
DetectStopSignal_Front(B1, B2)
Possibly more - maximum operations reached
```

☑	Status	Name
☑	✓	Rangierfahrt0_prob_fallstudie

```
MACHINE Rangierfahrt0_prob_fallstudie
EXTENDS Rangierfahrt0
DEFINITIONS
 VISB_JSON_FILE == "visb_track_fallstudie.json";
 CUSTOM_GRAPH_NODES == BLOCKS;
 CUSTOM_GRAPH_EDGES == next
CONSTANTS B347a, B347b, B347c, B855a, B855b, C55, Person
PROPERTIES
 OBJECTS = {lok, C55, Person} &
 card({lok, C55, Person}) = 3 &

 BLOCKS = {B347a, B347b, B347c, B855a, B855b} &
 card({B347a, B347b, B855a, B855b, B347c}) = 5 &

 TRK = {B347a |-> B347b,  B347b |-> B347c, B347a |-> B855a, B855a |-> B855b} &

 DERAILERS = {B855a |-> B855b, B347b |-> B347c} &

 SIGNALS = {B347a |-> B855a, B347a |-> B347b} \/ DERAILERS \/ DERAILERS~ &

 occ_init = {lok |-> B347a, C55 |-> B855b, Person |-> B855b} &
 next_init =  {B347a |-> B347b, B347b |-> B347c,
               B855a |-> B855b} &
 signals_init = {(B347a |-> B855a) |-> Sh0, (B347a |-> B347b) |-> Sh0} \/ (DEF

 block_length_init = BLOCKS * {100} &
 object_length_init = {lok |-> 30, C55 |-> 20, Person |-> 1} &
 object_front_init = {lok |-> ((BLOCKS-{B347a}) * {0} \/ {B347a |-> object_ler
                      C55 |-> ((BLOCKS-{B855b}) * {0} \/ {B855b |-> block_leng
                      Person |-> ((BLOCKS-{B855b}) * {0} \/ {B855b |-> block_l
 object_back_init = {lok |-> (BLOCKS * {0}),
                      C55 |-> ((BLOCKS-{B855b}) * {0} \/ {B855b |-> block_leng
                      Person |-> ((BLOCKS-{B855b}) * {0} \/ {B855b |-> block_l
END
```

Rangierfahrt0_prob_fallstudie
Rangierfahrt0_prob_fallstudie.mch

```
ProB B-Console
Model loaded: Rangierfahrt0_prob_fallstudie.mch
B>
```

```
0   ---root---
1   SETUP_CONSTANTS
2   INITIALISATION
3   MoveLokForwards(frnt=B34
4   MoveLokForwards(frnt=B34
5   MoveLokForwards(frnt=B34
6   MovePoint(Block=B347a, N
7   MoveLokForwards(frnt=B34
8   MoveLokForwards(frnt=B34
9   MoveLokForwards(frnt=B34
10  MoveLokForwards(frnt=B34
11  MoveLokForwards(frnt=B34
12  MoveLokForwards(frnt=B34
13  MoveLokForwards(frnt=B34
14  MoveLokForwards(frnt=B34
15  SwitchSignalToSh1(frnt=B34
16  MoveLokForwards(frnt=B34
```

Everything is OK

Models are available at:

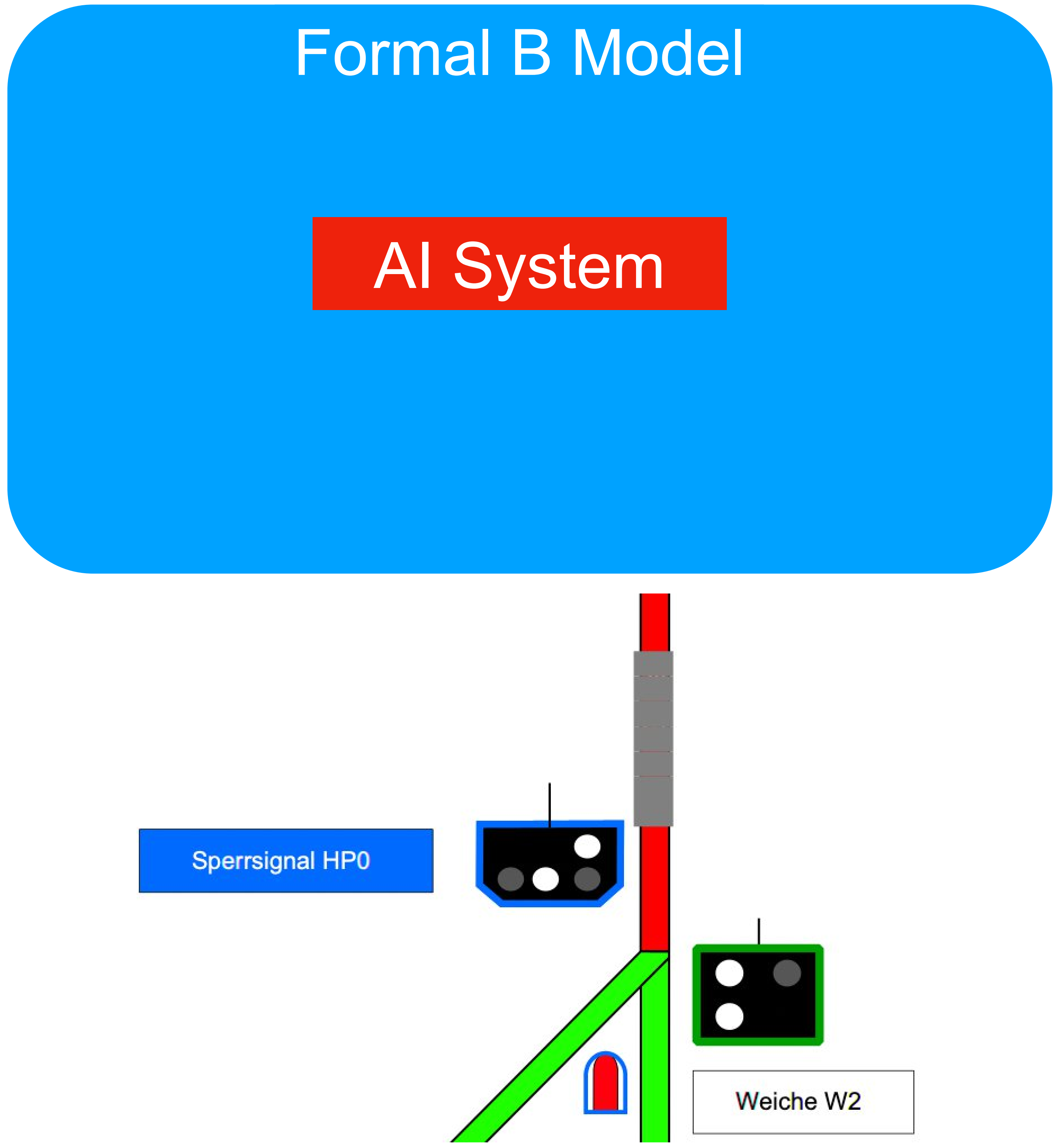
https://github.com/hhu-stups/kilok_shunting_model

Thank you for your attention!

APPENDIX

KI-LOK Formal Model

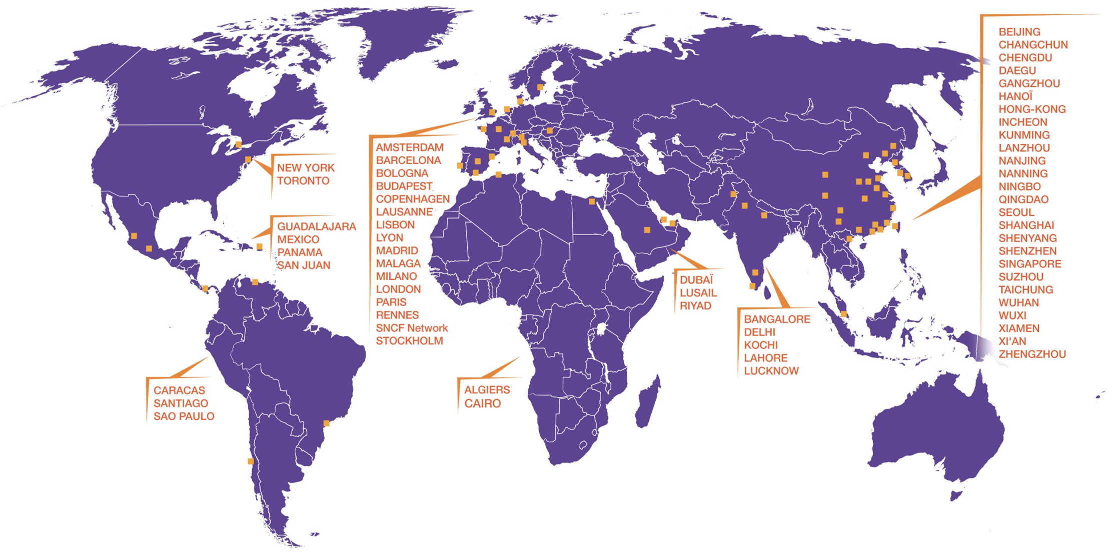
- Formal **B** system **model** of
  - Deterministic steering system of shunting movements
  - Environment (points, signals, derailers, obstacles)
  - AI: correct/false/non-detection of objects



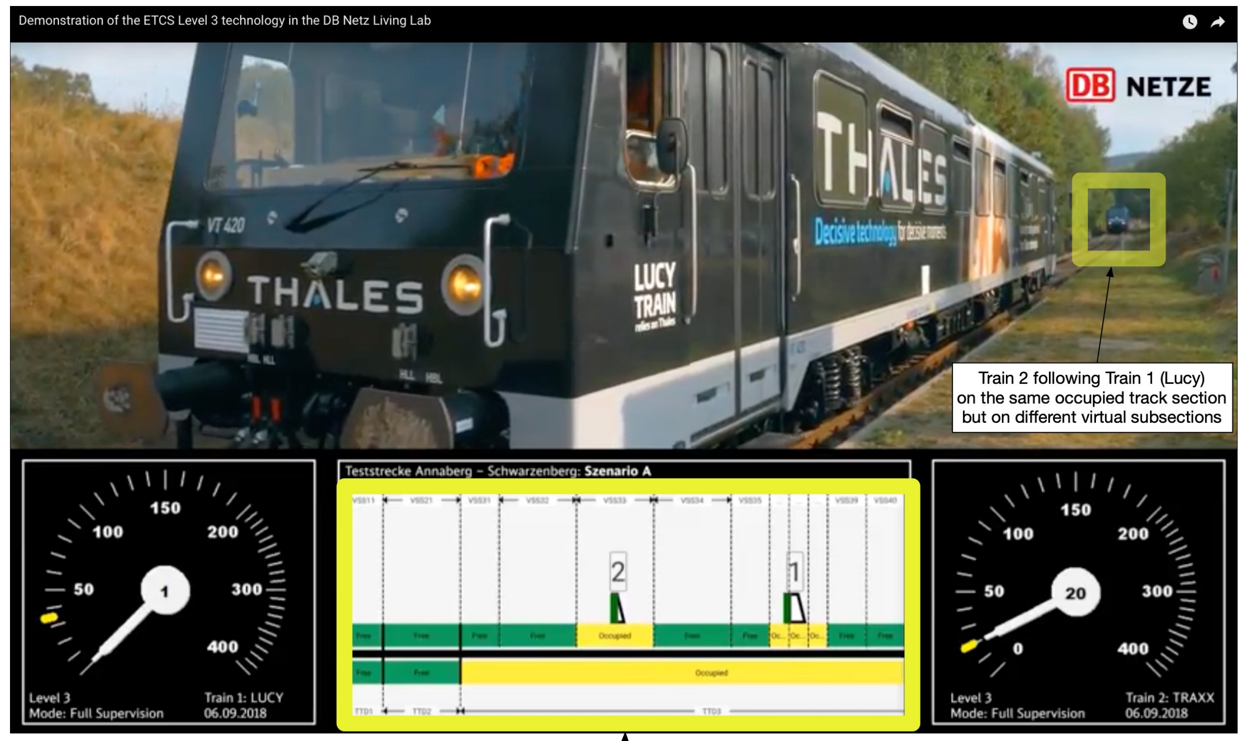
Side-note:
Industrial Usage of B and Event-B

Software
(30% of CBTC systems worldwide use B software)

Data & Config. Validation



System Specification & Executable Model



Demonstration of the ETCS Level 3 technology in the DB Netz Living Lab

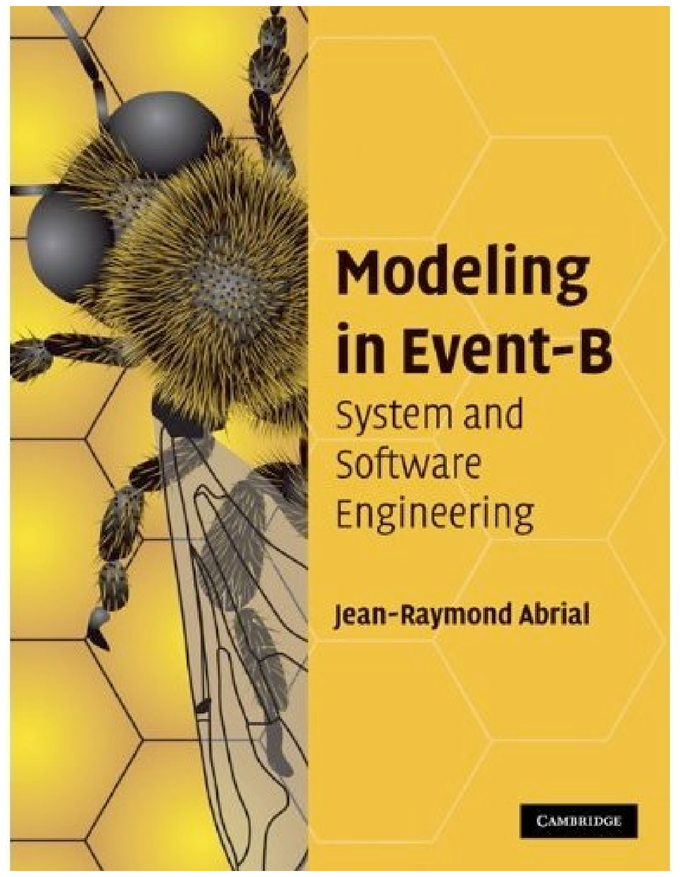
Train 2 following Train 1 (Lucy) on the same occupied track section but on different virtual subsections

Teststrecke Annaberg – Schwarzenberg: **Szenario A**

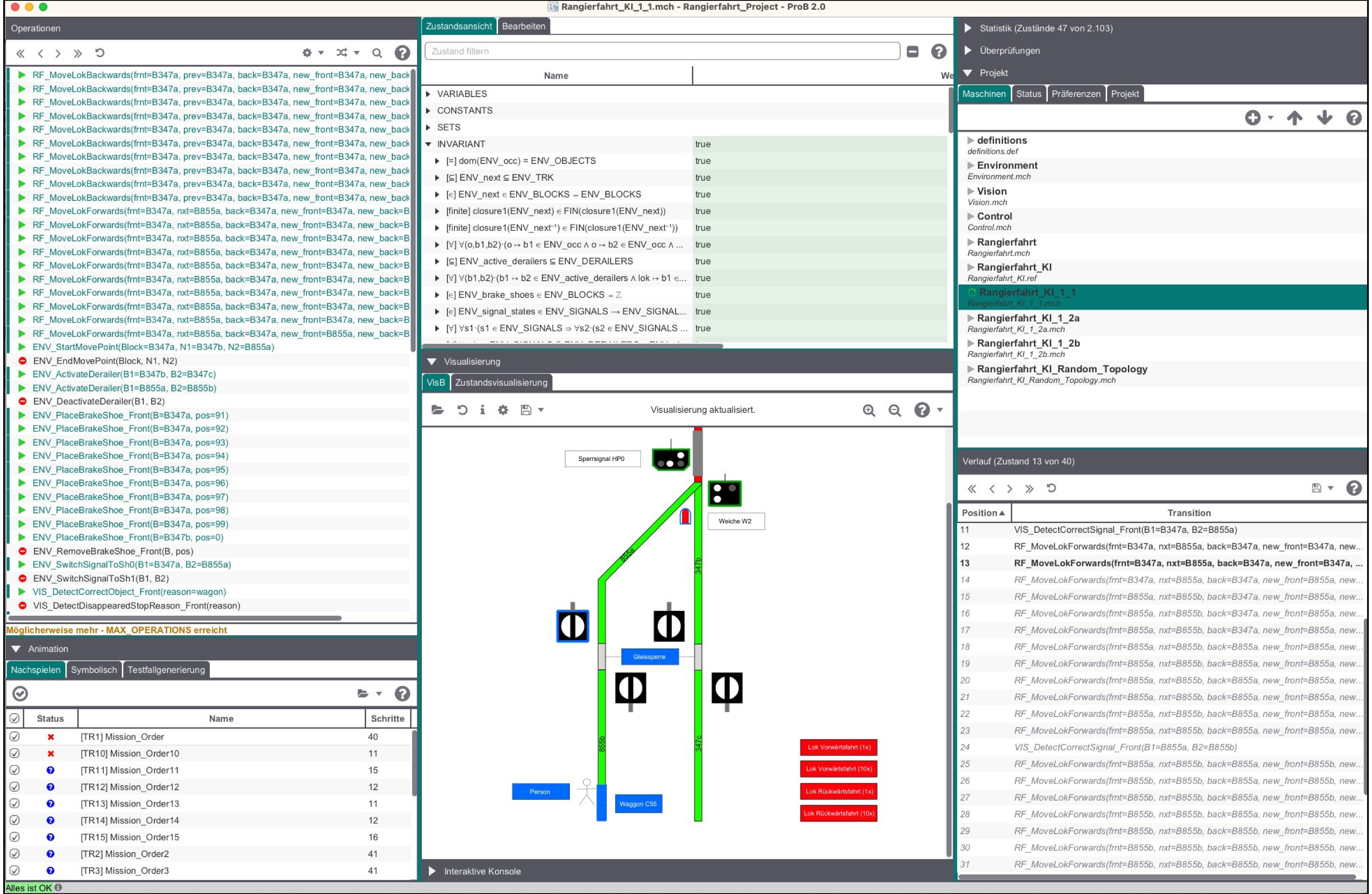
Level 3
Mode: Full Supervision
Train 1: LUCY
06.09.2018

Level 3
Mode: Full Supervision
Train 2: TRAXX
06.09.2018

ProB running in real-time animating a
formal B model of the **Hybrid-Level 3** principles
developed by a team from the University of Düsseldorf and Thales with support from ClearSy



System Analysis & Safety Case



Rangierfahrt_KI_1_1.mch - Rangierfahrt_Project - ProB 2.0

Operationen

```
RF_MoveLokBackwards(frnt=B347a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B347a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B347a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B347a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B347a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B347a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B347a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B347a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B347a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B347a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B347a, new_back=B
RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B347a, new_back=B
RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B347a, new_back=B
RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B347a, new_back=B
RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B347a, new_back=B
RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B347a, new_back=B
RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B347a, new_back=B
RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B347a, new_back=B
RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B347a, new_back=B
RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B855a, new_back=B
ENV_StartMovePoint(Block=B347a, N1=B347b, N2=B855a)
ENV_EndMovePoint(Block, N1, N2)
ENV_ActivateDerailer(B1=B347b, B2=B347c)
ENV_ActivateDerailer(B1=B855a, B2=B855b)
ENV_DeactivateDerailer(B1, B2)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=91)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=92)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=93)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=94)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=95)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=96)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=97)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=98)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=99)
ENV_PlaceBrakeShoe_Front(B=B347b, pos=0)
ENV_RemoveBrakeShoe_Front(B, pos)
ENV_SwitchSignalToSh0(B1=B347a, B2=B855a)
ENV_SwitchSignalToSh1(B1, B2)
VIS_DetectCorrectObject_Front(reason=wagon)
VIS_DetectDisappearedStopReason_Front(reason)
```

Möglicherweise mehr - MAX_OPERATIONS erreicht

Animation

Nachspielen | Symbolisch | Testfallgenerierung

✓	Status	Name	Schritte
✓	✗	[TR1] Mission_Order	40
✓	✗	[TR10] Mission_Order10	11
✓	?	[TR11] Mission_Order11	15
✓	?	[TR12] Mission_Order12	12
✓	?	[TR13] Mission_Order13	11
✓	?	[TR14] Mission_Order14	12
✓	?	[TR15] Mission_Order15	16
✓	?	[TR2] Mission_Order2	41
✓	?	[TR3] Mission_Order3	41

Alles ist OK

Zustandsansicht | Bearbeiten

Zustand filtern

Name	We
VARIABLES	
CONSTANTS	
SETS	
INVARIANT	true
[=] dom(ENV_occ) = ENV_OBJECTS	true
[⊆] ENV_next ⊆ ENV_TRK	true
[∈] ENV_next ∈ ENV_BLOCKS ⤔ ENV_BLOCKS	true
[finite] closure1(ENV_next) ∈ FIN(closure1(ENV_next))	true
[finite] closure1(ENV_next⁻¹) ∈ FIN(closure1(ENV_next⁻¹))	true
[∀] ∀(o,b1,b2)·(o ↦ b1 ∈ ENV_occ ∧ o ↦ b2 ∈ ENV_occ ∧ ...	true
[⊆] ENV_active_derailers ⊆ ENV_DERAILERS	true
[∀] ∀(b1,b2)·(b1 ↦ b2 ∈ ENV_active_derailers ∧ lok ↦ b1 ∈...	true
[∈] ENV_brake_shoes ∈ ENV_BLOCKS ⇸ ℤ	true
[∈] ENV_signal_states ∈ ENV_SIGNALS → ENV_SIGNAL...	true
[∀] ∀s1·(s1 ∈ ENV_SIGNALS ⇒ ∀s2·(s2 ∈ ENV_SIGNALS ...	true

Visualisierung

VisB | Zustandsvisualisierung

Visualisierung aktualisiert.

Interaktive Konsole

Statistik (Zustände 47 von 2.103)

Überprüfungen

Projekt

Maschinen | Status | Präferenzen | Projekt

definitions
definitions.def

Environment
Environment.mch

Vision
Vision.mch

Control
Control.mch

Rangierfahrt
Rangierfahrt.mch

Rangierfahrt_KI
Rangierfahrt_KI.ref

Rangierfahrt_KI_1_1
Rangierfahrt_KI_1_1.mch

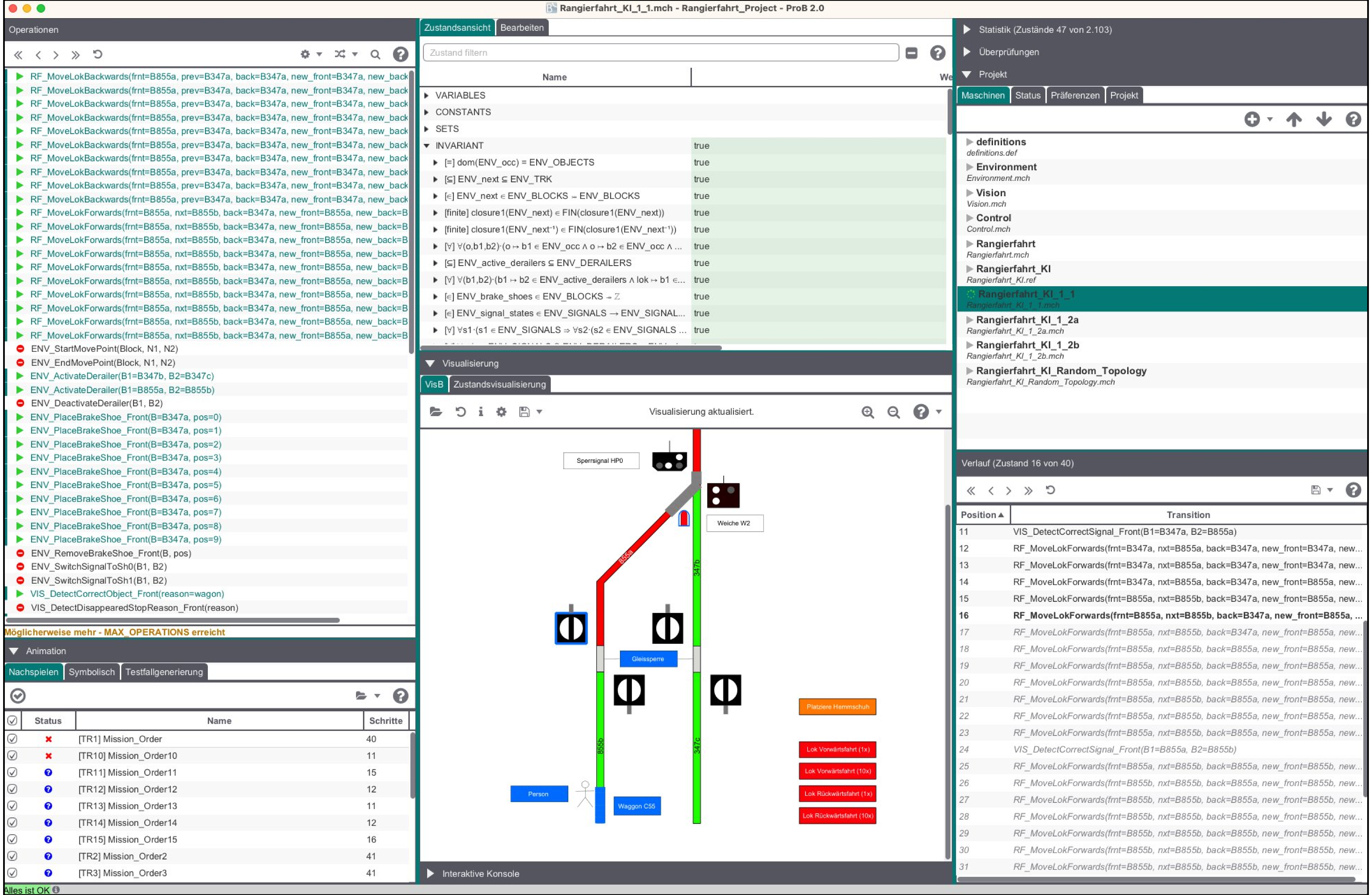
Rangierfahrt_KI_1_2a
Rangierfahrt_KI_1_2a.mch

Rangierfahrt_KI_1_2b
Rangierfahrt_KI_1_2b.mch

Rangierfahrt_KI_Random_Topology
Rangierfahrt_KI_Random_Topology.mch

Verlauf (Zustand 13 von 40)

Position ▲	Transition
11	VIS_DetectCorrectSignal_Front(B1=B347a, B2=B855a)
12	RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B347a, new...
13	**RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B347a, ...**
14	*RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B855a, new...*
15	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B347a, new_front=B855a, new...*
16	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B347a, new_front=B855a, new...*
17	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B347a, new_front=B855a, new...*
18	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new...*
19	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new...*
20	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new...*
21	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new...*
22	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new...*
23	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new...*
24	*VIS_DetectCorrectSignal_Front(B1=B855a, B2=B855b)*
25	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855b, new...*
26	*RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855a, new_front=B855b, new...*
27	*RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855a, new_front=B855b, new...*
28	*RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855a, new_front=B855b, new...*
29	*RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_front=B855b, new...*
30	*RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_front=B855b, new...*
31	*RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_front=B855b, new...*



Rangierfahrt_KI_1_1.mch - Rangierfahrt_Project - ProB 2.0

Operationen

```
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B347a, new_front=B347a, new_back
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B347a, new_front=B855a, new_back=B
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B347a, new_front=B855a, new_back=B
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B347a, new_front=B855a, new_back=B
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B347a, new_front=B855a, new_back=B
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B347a, new_front=B855a, new_back=B
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B347a, new_front=B855a, new_back=B
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B347a, new_front=B855a, new_back=B
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B347a, new_front=B855a, new_back=B
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B347a, new_front=B855a, new_back=B
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B347a, new_front=B855a, new_back=B
ENV_StartMovePoint(Block, N1, N2)
ENV_EndMovePoint(Block, N1, N2)
ENV_ActivateDerailer(B1=B347b, B2=B347c)
ENV_ActivateDerailer(B1=B855a, B2=B855b)
ENV_DeactivateDerailer(B1, B2)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=0)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=1)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=2)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=3)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=4)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=5)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=6)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=7)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=8)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=9)
ENV_RemoveBrakeShoe_Front(B, pos)
ENV_SwitchSignalToSh0(B1, B2)
ENV_SwitchSignalToSh1(B1, B2)
VIS_DetectCorrectObject_Front(reason=wagon)
VIS_DetectDisappearedStopReason_Front(reason)
```

Möglicherweise mehr - MAX_OPERATIONS erreicht

Animation

Nachspielen | Symbolisch | Testfallgenerierung

☑	Status	Name	Schritte
☑	✗	[TR1] Mission_Order	40
☑	✗	[TR10] Mission_Order10	11
☑	?	[TR11] Mission_Order11	15
☑	?	[TR12] Mission_Order12	12
☑	?	[TR13] Mission_Order13	11
☑	?	[TR14] Mission_Order14	12
☑	?	[TR15] Mission_Order15	16
☑	?	[TR2] Mission_Order2	41
☑	?	[TR3] Mission_Order3	41

Alles ist OK

Zustandsansicht | Bearbeiten

Zustand filtern

Name	We
VARIABLES	
CONSTANTS	
SETS	
INVARIANT	true
[=] dom(ENV_occ) = ENV_OBJECTS	true
[⊆] ENV_next ⊆ ENV_TRK	true
[∈] ENV_next ∈ ENV_BLOCKS ⤔ ENV_BLOCKS	true
[finite] closure1(ENV_next) ∈ FIN(closure1(ENV_next))	true
[finite] closure1(ENV_next⁻¹) ∈ FIN(closure1(ENV_next⁻¹))	true
[∀] ∀(o,b1,b2)·(o ↦ b1 ∈ ENV_occ ∧ o ↦ b2 ∈ ENV_occ ∧ ...	true
[⊆] ENV_active_derailers ⊆ ENV_DERAILERS	true
[∀] ∀(b1,b2)·(b1 ↦ b2 ∈ ENV_active_derailers ∧ lok ↦ b1 ∈...	true
[∈] ENV_brake_shoes ∈ ENV_BLOCKS ⇸ ℤ	true
[∈] ENV_signal_states ∈ ENV_SIGNALS → ENV_SIGNAL...	true
[∀] ∀s1·(s1 ∈ ENV_SIGNALS ⇒ ∀s2·(s2 ∈ ENV_SIGNALS ...	true

Visualisierung

VisB | Zustandsvisualisierung

Visualisierung aktualisiert.

Interaktive Konsole

Statistik (Zustände 47 von 2.103)

Überprüfungen

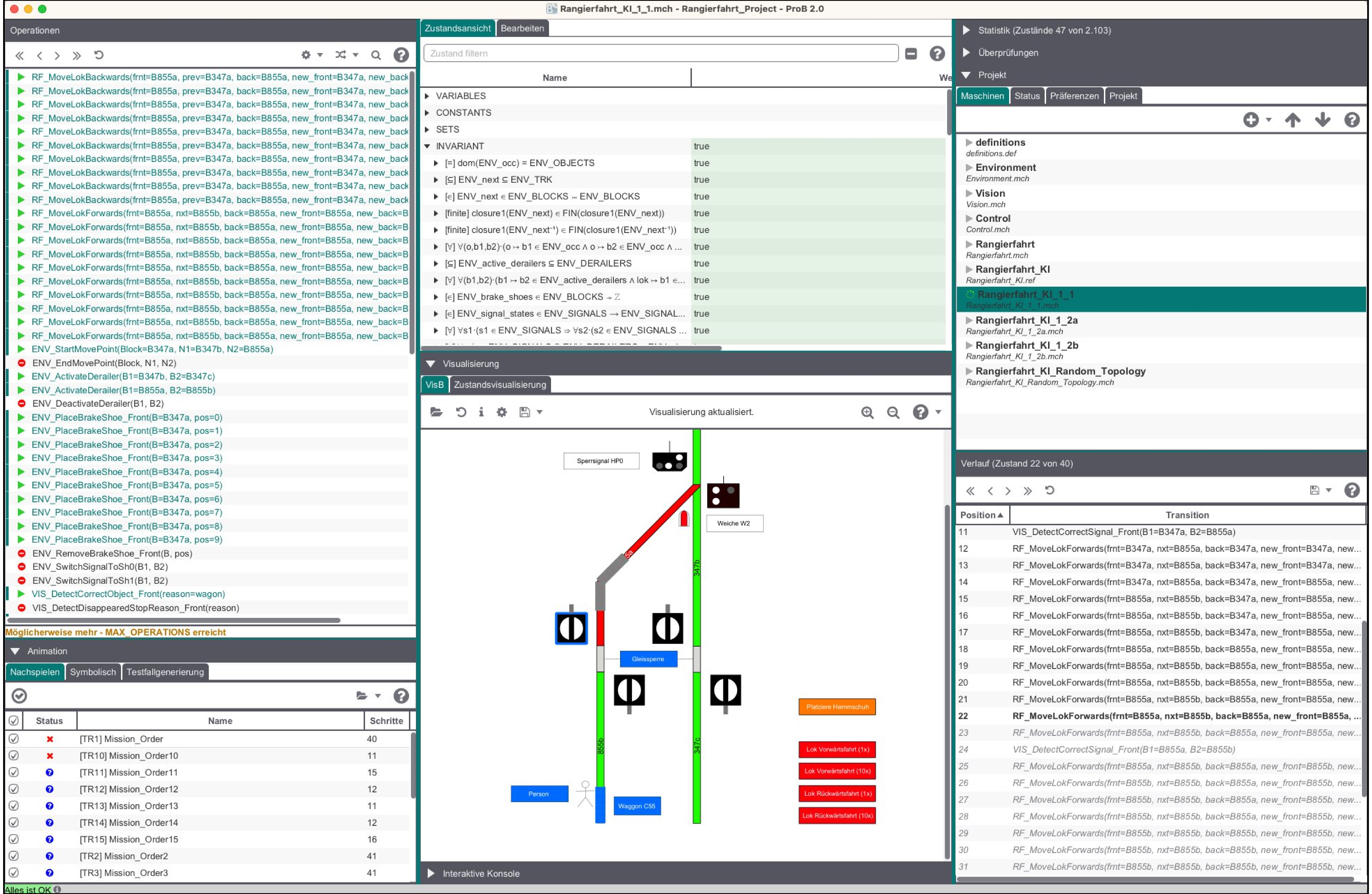
Projekt

Maschinen | Status | Präferenzen | Projekt

- **definitions** *definitions.def*
- **Environment** *Environment.mch*
- **Vision** *Vision.mch*
- **Control** *Control.mch*
- **Rangierfahrt** *Rangierfahrt.mch*
- **Rangierfahrt_KI** *Rangierfahrt_KI.ref*
- **Rangierfahrt_KI_1_1** *Rangierfahrt_KI_1_1.mch*
- **Rangierfahrt_KI_1_2a** *Rangierfahrt_KI_1_2a.mch*
- **Rangierfahrt_KI_1_2b** *Rangierfahrt_KI_1_2b.mch*
- **Rangierfahrt_KI_Random_Topology** *Rangierfahrt_KI_Random_Topology.mch*

Verlauf (Zustand 16 von 40)

Position ▲	Transition
11	VIS_DetectCorrectSignal_Front(B1=B347a, B2=B855a)
12	RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B347a, new...
13	RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B347a, new...
14	RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B855a, new...
15	RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B347a, new_front=B855a, new...
16	**RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B347a, new_front=B855a, ...**
17	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B347a, new_front=B855a, new...*
18	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new...*
19	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new...*
20	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new...*
21	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new...*
22	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new...*
23	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new...*
24	*VIS_DetectCorrectSignal_Front(B1=B855a, B2=B855b)*
25	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855b, new...*
26	*RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855a, new_front=B855b, new...*
27	*RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855a, new_front=B855b, new...*
28	*RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855a, new_front=B855b, new...*
29	*RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_front=B855b, new...*
30	*RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_front=B855b, new...*
31	*RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_front=B855b, new...*



Rangierfahrt_KI_1_1.mch - Rangierfahrt_Project - ProB 2.0

Operationen

```
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B855a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B855a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B855a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B855a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B855a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B855a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B855a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B855a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B855a, new_front=B347a, new_back
RF_MoveLokBackwards(frnt=B855a, prev=B347a, back=B855a, new_front=B347a, new_back
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new_back=B
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new_back=B
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new_back=B
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new_back=B
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new_back=B
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new_back=B
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new_back=B
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new_back=B
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new_back=B
RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new_back=B
ENV_StartMovePoint(Block=B347a, N1=B347b, N2=B855a)
ENV_EndMovePoint(Block, N1, N2)
ENV_ActivateDerailer(B1=B347b, B2=B347c)
ENV_ActivateDerailer(B1=B855a, B2=B855b)
ENV_DeactivateDerailer(B1, B2)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=0)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=1)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=2)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=3)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=4)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=5)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=6)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=7)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=8)
ENV_PlaceBrakeShoe_Front(B=B347a, pos=9)
ENV_RemoveBrakeShoe_Front(B, pos)
ENV_SwitchSignalToSh0(B1, B2)
ENV_SwitchSignalToSh1(B1, B2)
VIS_DetectCorrectObject_Front(reason=wagon)
VIS_DetectDisappearedStopReason_Front(reason)
```

Möglicherweise mehr - MAX_OPERATIONS erreicht

Animation

Nachspielen | Symbolisch | Testfallgenerierung

☑	Status	Name	Schritte
☑	✗	[TR1] Mission_Order	40
☑	✗	[TR10] Mission_Order10	11
☑	?	[TR11] Mission_Order11	15
☑	?	[TR12] Mission_Order12	12
☑	?	[TR13] Mission_Order13	11
☑	?	[TR14] Mission_Order14	12
☑	?	[TR15] Mission_Order15	16
☑	?	[TR2] Mission_Order2	41
☑	?	[TR3] Mission_Order3	41

Alles ist OK

Zustandsansicht | Bearbeiten

Zustand filtern

Name	We
VARIABLES	
CONSTANTS	
SETS	
INVARIANT	true
[=] dom(ENV_occ) = ENV_OBJECTS	true
[⊆] ENV_next ⊆ ENV_TRK	true
[∈] ENV_next ∈ ENV_BLOCKS ⤔ ENV_BLOCKS	true
[finite] closure1(ENV_next) ∈ FIN(closure1(ENV_next))	true
[finite] closure1(ENV_next⁻¹) ∈ FIN(closure1(ENV_next⁻¹))	true
[∀] ∀(o,b1,b2)·(o ↦ b1 ∈ ENV_occ ∧ o ↦ b2 ∈ ENV_occ ∧ ...	true
[⊆] ENV_active_derailers ⊆ ENV_DERAILERS	true
[∀] ∀(b1,b2)·(b1 ↦ b2 ∈ ENV_active_derailers ∧ lok ↦ b1 ∈...	true
[∈] ENV_brake_shoes ∈ ENV_BLOCKS ⇸ ℤ	true
[∈] ENV_signal_states ∈ ENV_SIGNALS → ENV_SIGNAL...	true
[∀] ∀s1·(s1 ∈ ENV_SIGNALS ⇒ ∀s2·(s2 ∈ ENV_SIGNALS ...	true

Visualisierung

VisB | Zustandsvisualisierung

Visualisierung aktualisiert.

Interaktive Konsole

Statistik (Zustände 47 von 2.103)

Überprüfungen

Projekt

Maschinen | Status | Präferenzen | Projekt

definitions
definitions.def

Environment
Environment.mch

Vision
Vision.mch

Control
Control.mch

Rangierfahrt
Rangierfahrt.mch

Rangierfahrt_KI
Rangierfahrt_KI.ref

Rangierfahrt_KI_1_1
Rangierfahrt_KI_1_1.mch

Rangierfahrt_KI_1_2a
Rangierfahrt_KI_1_2a.mch

Rangierfahrt_KI_1_2b
Rangierfahrt_KI_1_2b.mch

Rangierfahrt_KI_Random_Topology
Rangierfahrt_KI_Random_Topology.mch

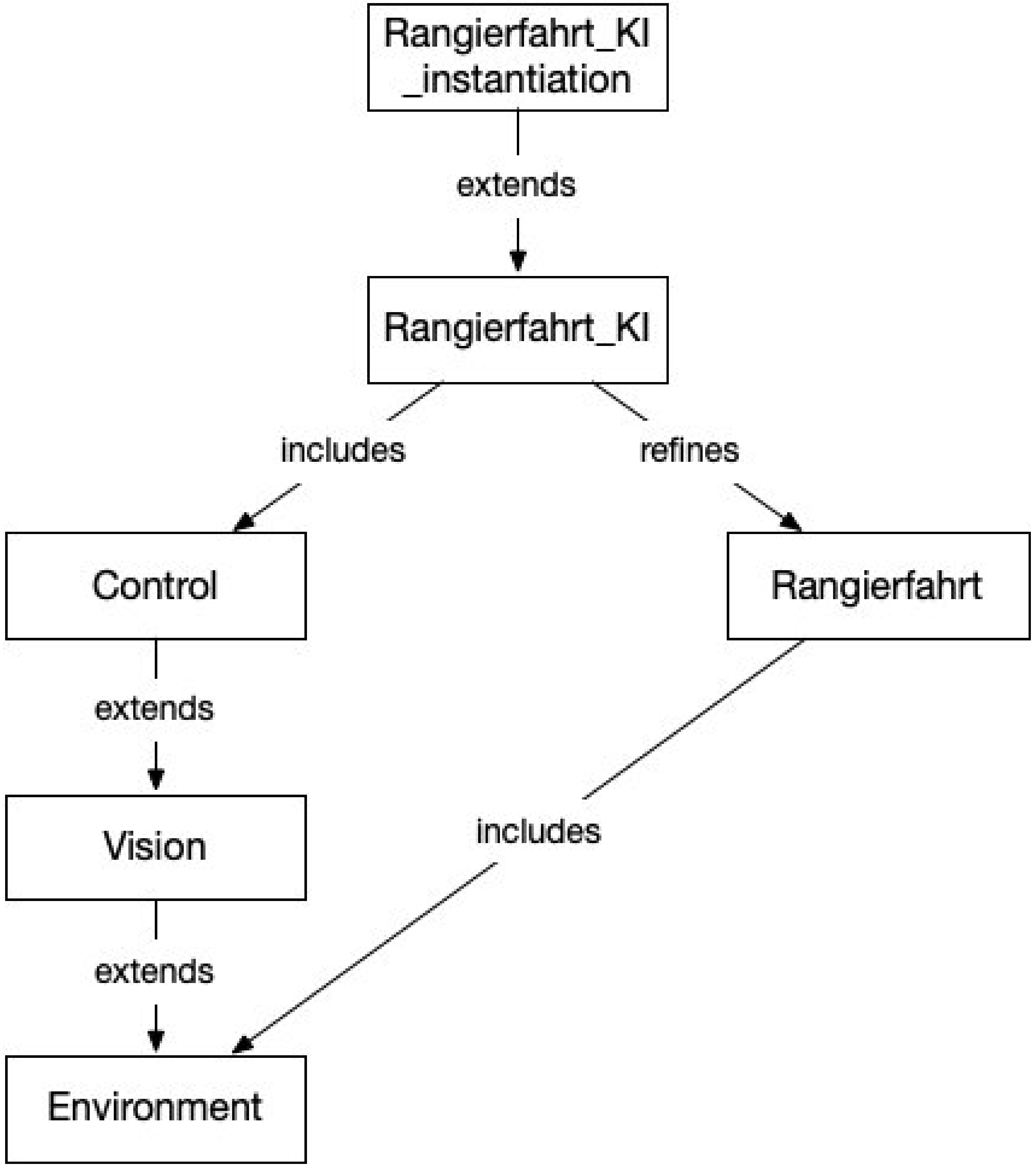
Verlauf (Zustand 22 von 40)

Position ▲	Transition
11	VIS_DetectCorrectSignal_Front(B1=B347a, B2=B855a)
12	RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B347a, new...
13	RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B347a, new...
14	RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B855a, new...
15	RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B347a, new_front=B855a, new...
16	RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B347a, new_front=B855a, new...
17	RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B347a, new_front=B855a, new...
18	RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new...
19	RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new...
20	RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new...
21	RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new...
22	**RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, ...**
23	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new...*
24	*VIS_DetectCorrectSignal_Front(B1=B855a, B2=B855b)*
25	*RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855b, new...*
26	*RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855a, new_front=B855b, new...*
27	*RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855a, new_front=B855b, new...*
28	*RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_front=B855b, new...*
29	*RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_front=B855b, new...*
30	*RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_front=B855b, new...*
31	*RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_front=B855b, new...*

Approaches in this Work

- Certified Control - for V&V of perception system
- Formal Methods - for V&V of steering system

Machine Hierarchy



Mission Order

Drive from the current position on track 347 to position B on track 855.
Position B is defined as wagon C55’s position (QR code).
Approach the wagon to the clutch position.
Recognise all field elements and people.
The task for the system: Recognise the described field elements (points, derailers, brake shoes) and signals reliably.

Mission Order

1. Drive from the current position on track 347a to the stop signal and point
2. Recognise stop signal and point position
3. Enter 855a and drive to the derailer
4. Recognise derailer
5. Enter 855b and approach wagon to the clutch position
6. Recognise the person and the wagon

Steps (2), (4) and (6) must recognise field elements or people correctly, otherwise, the Mission Order might not be achieved.

Mission Order

- Validation by 24 traces with different variations:

  - Neither wagon nor person recognised correctly — leads to collision with both
  - Person recognised correctly, but not wagon — leads to collision with wagon
  - Active derailer not recognised correctly — leads to the train entering a section where collision is possible
  - Neither stop signal nor moving point position recognised correctly — leads to the train derailing
  - Point position recognised correctly, but not stop signal — leads to the train entering a section where collision is possible

Safety Properties: SAF1 – SAF5

SAF1-5: When point positions, stop signals, derailers, and obstacles are recognised correctly, the train must not enter a safety-critical state (train derailing, train entering a blocked session, or collision with an obstacle).

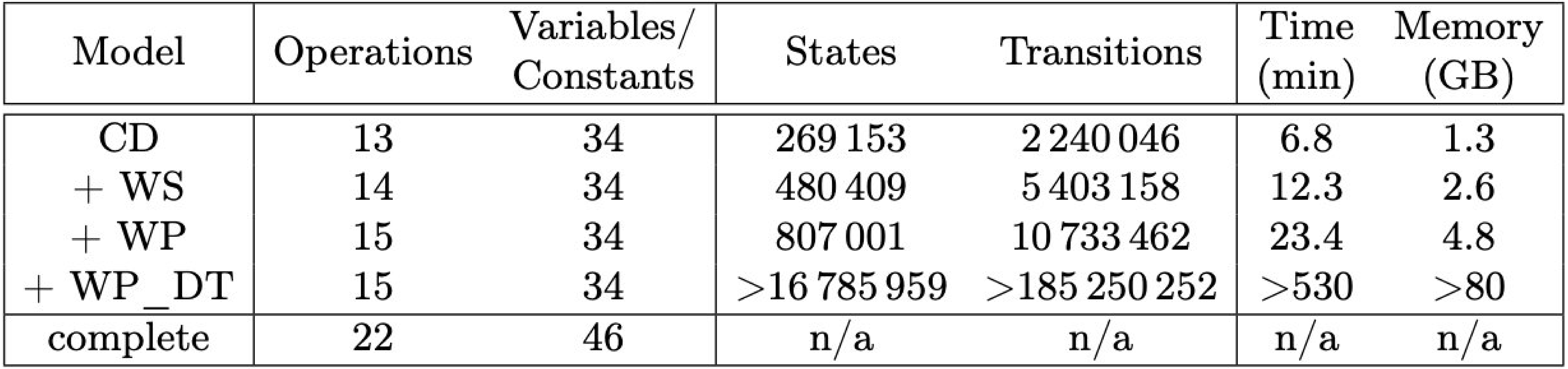
Validation by LTL Formula:

G({“train moves forwards” ⇒
  Y (“control unit updates decision to move train forwards” ∧
      “train detected all signals correctly” ∧
      “train detected points correctly”∧
      “train detected obstacles correctly” ∧
      “train detected track correctly”})
⇒ G({“train does not reach safety-critical situation”})

Model Checking

- CD = Correct Detection, WS = Wrong Signals, WP = Wrong Detection of Points, WP_DT = Update detected track

Table 2: Model Checking Results for Selected Reduced Models



Model	Operations	Variables/ Constants	States	Transitions	Time (min)	Memory (GB)
CD	13	34	269 153	2 240 046	6.8	1.3
+ WS	14	34	480 409	5 403 158	12.3	2.6
+ WP	15	34	807 001	10 733 462	23.4	4.8
+ WP_DT	15	34	>16 785 959	>185 250 252	>530	>80
complete	22	46	n/a	n/a	n/a	n/a

Simulation with SimB

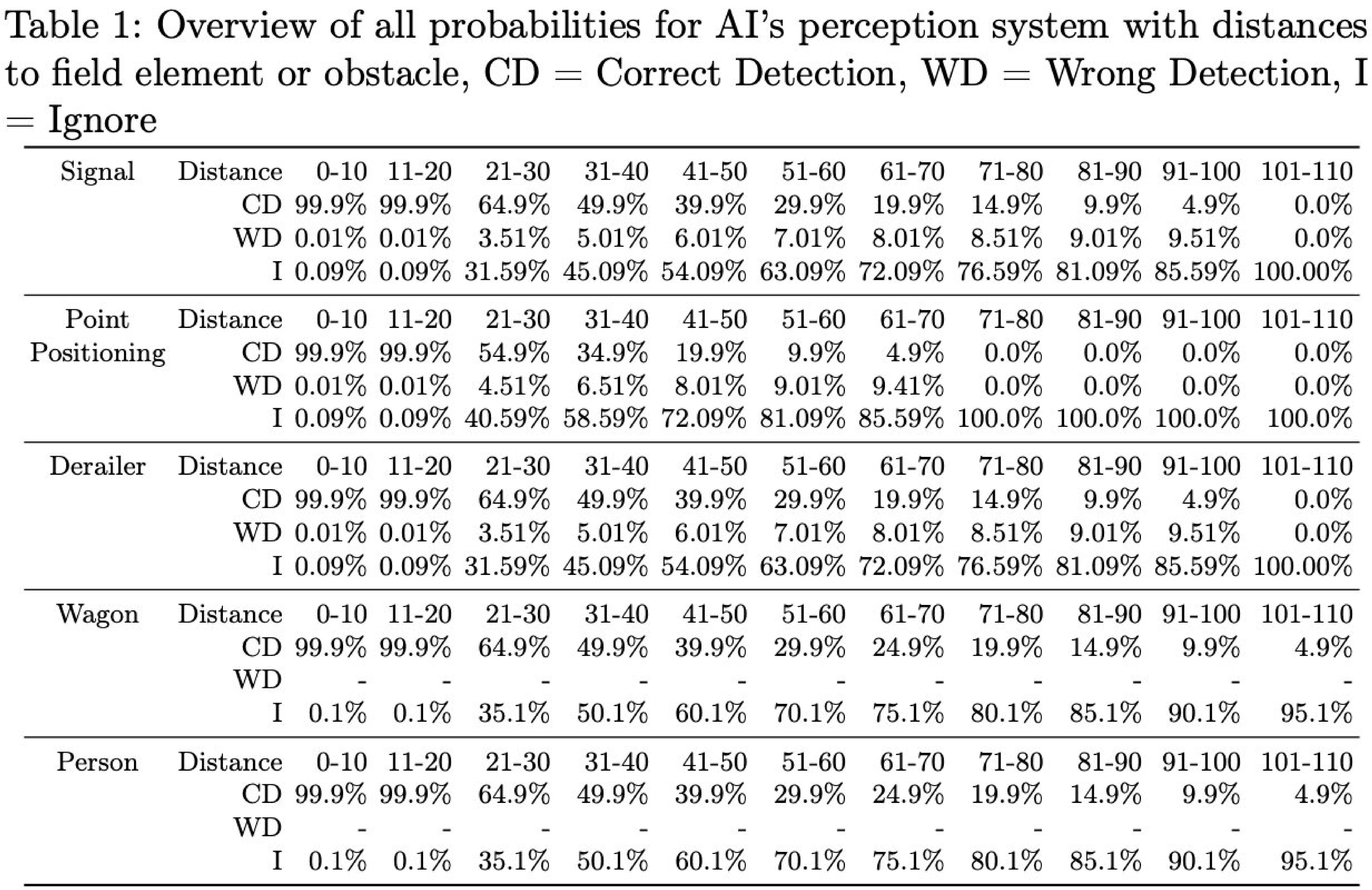


Table 1: Overview of all probabilities for AI's perception system with distances to field element or obstacle, CD = Correct Detection, WD = Wrong Detection, I = Ignore

Signal	Distance	0-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80	81-90	91-100	101-110
	CD	99.9%	99.9%	64.9%	49.9%	39.9%	29.9%	19.9%	14.9%	9.9%	4.9%	0.0%
	WD	0.01%	0.01%	3.51%	5.01%	6.01%	7.01%	8.01%	8.51%	9.01%	9.51%	0.0%
	I	0.09%	0.09%	31.59%	45.09%	54.09%	63.09%	72.09%	76.59%	81.09%	85.59%	100.00%
Point Positioning	Distance	0-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80	81-90	91-100	101-110
	CD	99.9%	99.9%	54.9%	34.9%	19.9%	9.9%	4.9%	0.0%	0.0%	0.0%	0.0%
	WD	0.01%	0.01%	4.51%	6.51%	8.01%	9.01%	9.41%	0.0%	0.0%	0.0%	0.0%
	I	0.09%	0.09%	40.59%	58.59%	72.09%	81.09%	85.59%	100.0%	100.0%	100.0%	100.0%
Derailer	Distance	0-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80	81-90	91-100	101-110
	CD	99.9%	99.9%	64.9%	49.9%	39.9%	29.9%	19.9%	14.9%	9.9%	4.9%	0.0%
	WD	0.01%	0.01%	3.51%	5.01%	6.01%	7.01%	8.01%	8.51%	9.01%	9.51%	0.0%
	I	0.09%	0.09%	31.59%	45.09%	54.09%	63.09%	72.09%	76.59%	81.09%	85.59%	100.00%
Wagon	Distance	0-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80	81-90	91-100	101-110
	CD	99.9%	99.9%	64.9%	49.9%	39.9%	29.9%	24.9%	19.9%	14.9%	9.9%	4.9%
	WD	-	-	-	-	-	-	-	-	-	-	-
	I	0.1%	0.1%	35.1%	50.1%	60.1%	70.1%	75.1%	80.1%	85.1%	90.1%	95.1%
Person	Distance	0-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80	81-90	91-100	101-110
	CD	99.9%	99.9%	64.9%	49.9%	39.9%	29.9%	24.9%	19.9%	14.9%	9.9%	4.9%
	WD	-	-	-	-	-	-	-	-	-	-	-
	I	0.1%	0.1%	35.1%	50.1%	60.1%	70.1%	75.1%	80.1%	85.1%	90.1%	95.1%

Probabilistic Property: PROP1

PROP1: When driving along the route from 347a to 855b, safety-critical situations (train derailing, train entering a blocked section, collision with wagon or person) must occur less frequently with KI-LOK than with humans

Validation by Simulation:

SIM(*ending*: “train reaches the end of 855b” ∨
“train reaches the end of 347c” ∨
“train reaches a safety-critical situation”
prop: “train never reaches a safety-critical situation”
check: HYPOTHESIS
procedure: LEFT_TAILED
probability: 0.999
α: 0.001)

Probabilistic Property: PROP2

PROP2: The probability of achieving the mission order by KI-LOK must be as good as humans

Validation by Simulation:

SIM(*ending*: “train reaches the end of 855b” ∨
“train reaches the end of 347c” ∨
“train reaches a safety-critical situation”
prop: “train reaches the end of 855b safely”
check: HYPOTHESIS
procedure: LEFT_TAILED
probability: 0.999
α: 0.001)

Validation and Verification: Challenges/Problems

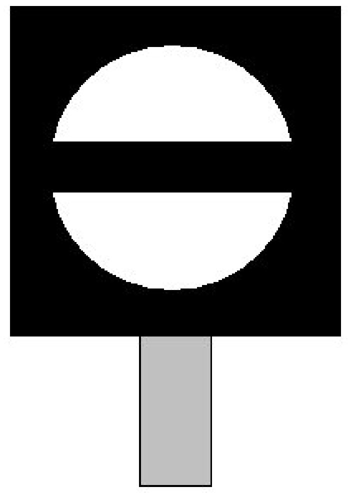
- Validation:
  - Traces only cover parts of state space
  - Probabilities for simulation

- Verification:
  - Model checking struggles with state space explosion
  - Proving very hard on our model
  - Perception System (**REC1-REC5**) - hard to verify with formal methods - use **certificate checking**

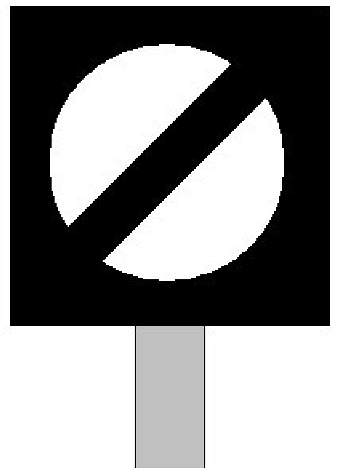
Sign Detection + Certificate Checker

Our Implementation

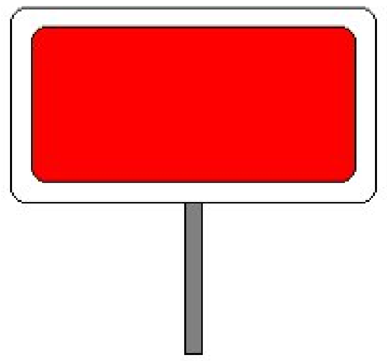
- Various Yolo v8 models trained for shunting signs
- Certificate: bounding box and class of detected sign
- Certificate checker
  - hand-written using OpenCV
  - feature detection + rules



Signal Sh0
STOP!



Signal Sh1
Shunting permitted



Signal Sh2
Protective STOP

etc.

Related Work

- New standardisation approaches, e.g., UL4600 standard

- Several approaches for verifying neural networks - do not scale for our properties and network

- Several railway systems: Abrial’s interlocking model, CBTC, Hybrid Level 3, ... - especially older railway systems - do not cover AI aspect