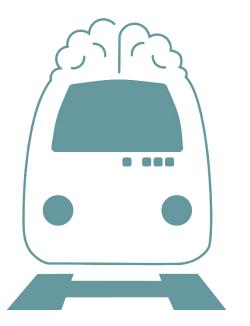


A Formal Model of Train Control with **Al-based Obstacle Detection** RSSR'2023, Berlin

Jan Gruteser, David Geleßus, Michael Leuschel, Jan Roßbach, Fabian Vu 12.10.2023

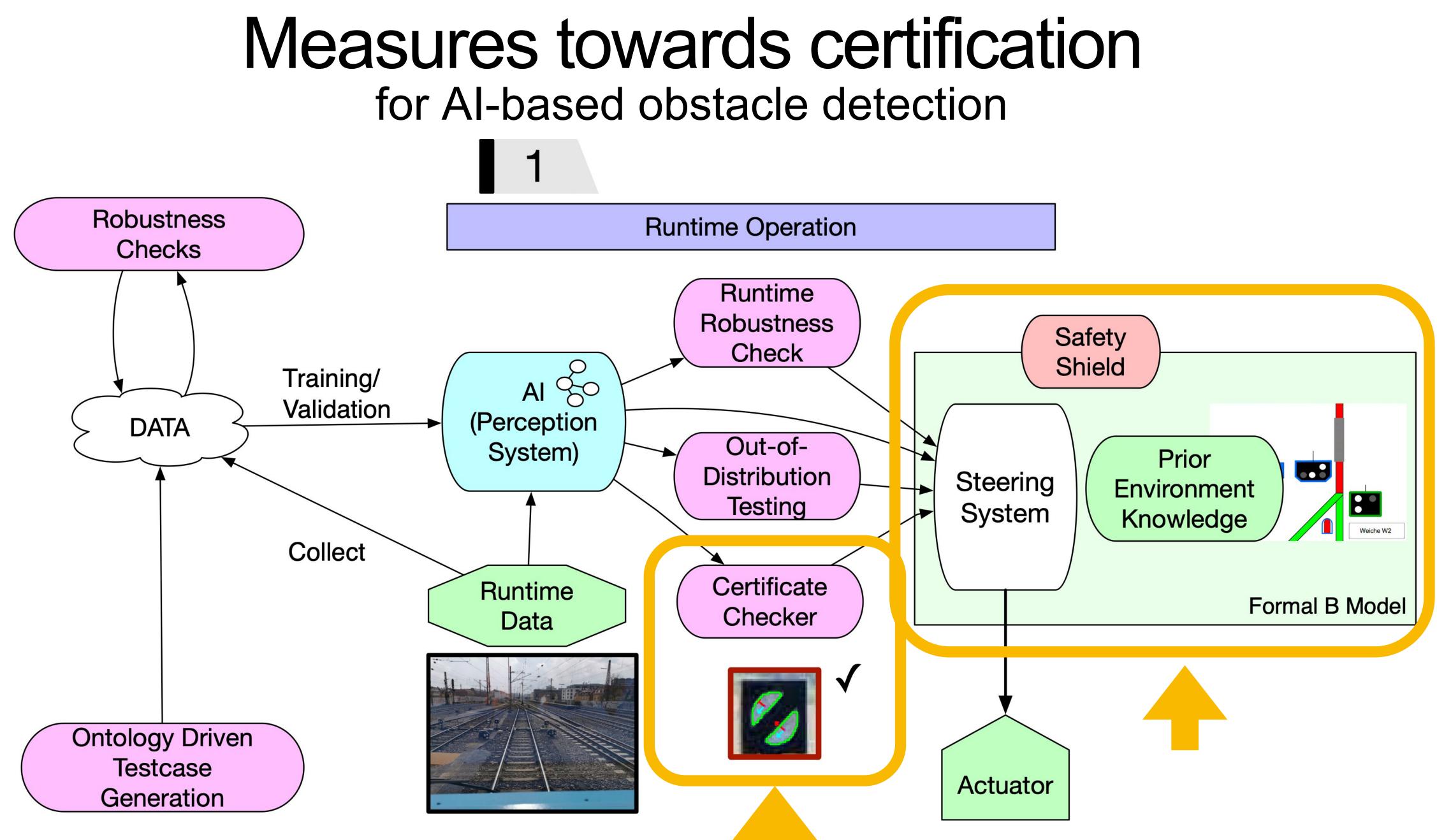


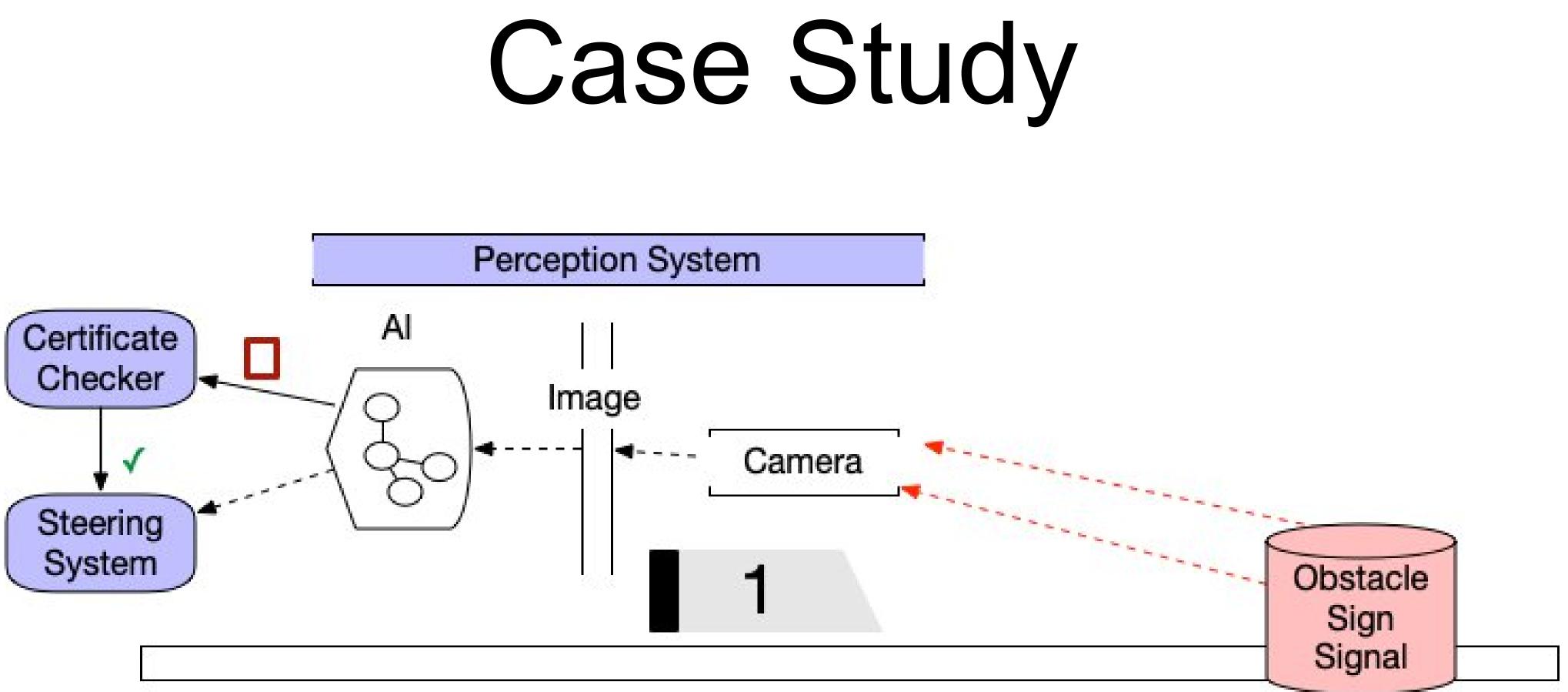


Gefördert durch: Bundesministerium für Wirtschaf und Energi aufgrund eines Beschlusses des Deutschen Bundestages

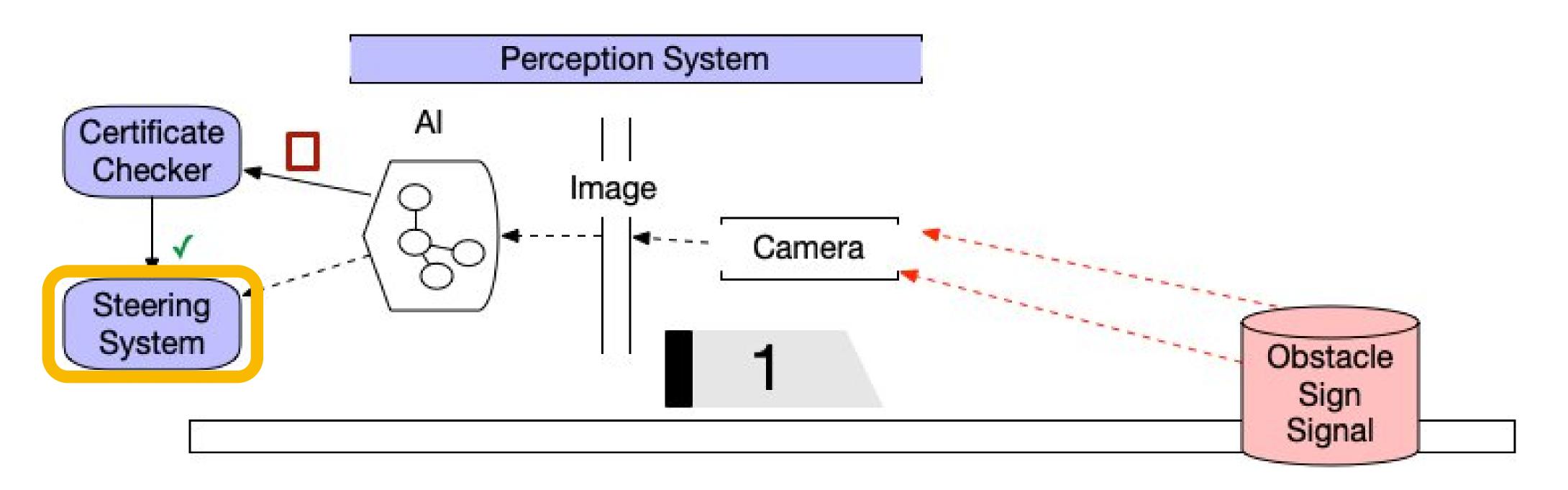
Motivation

- Al 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

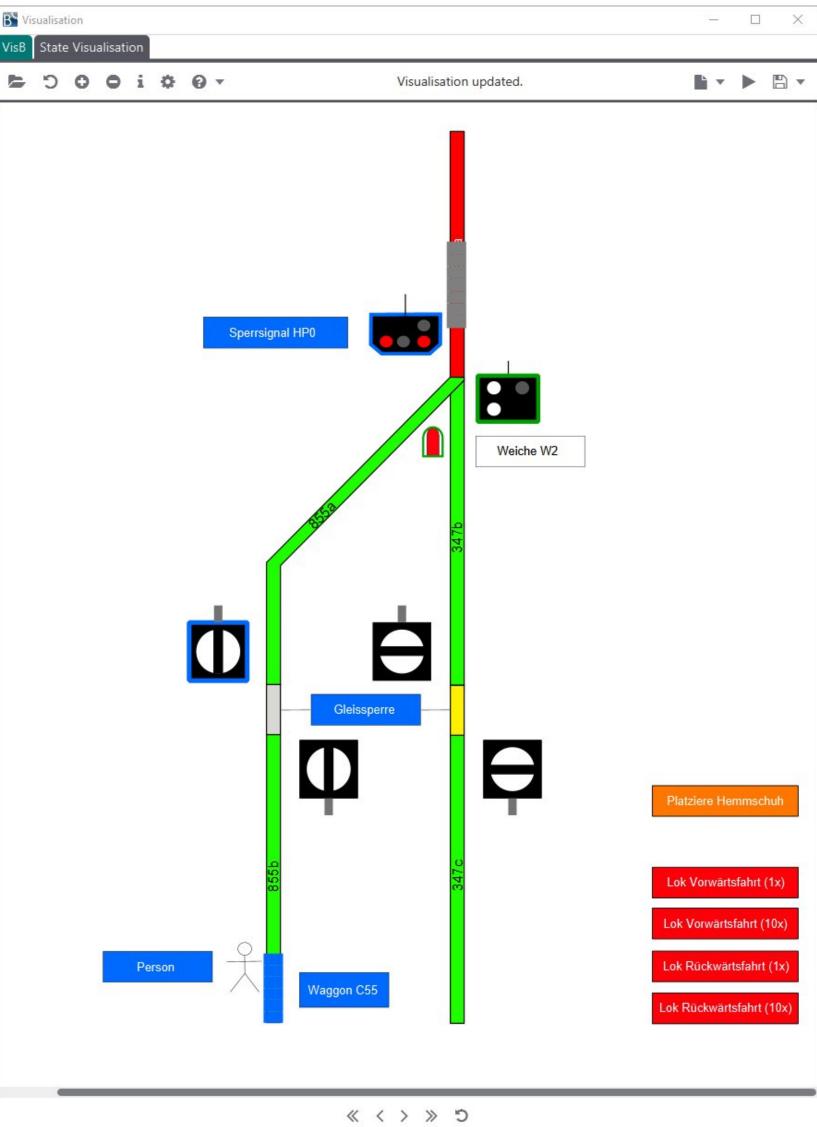




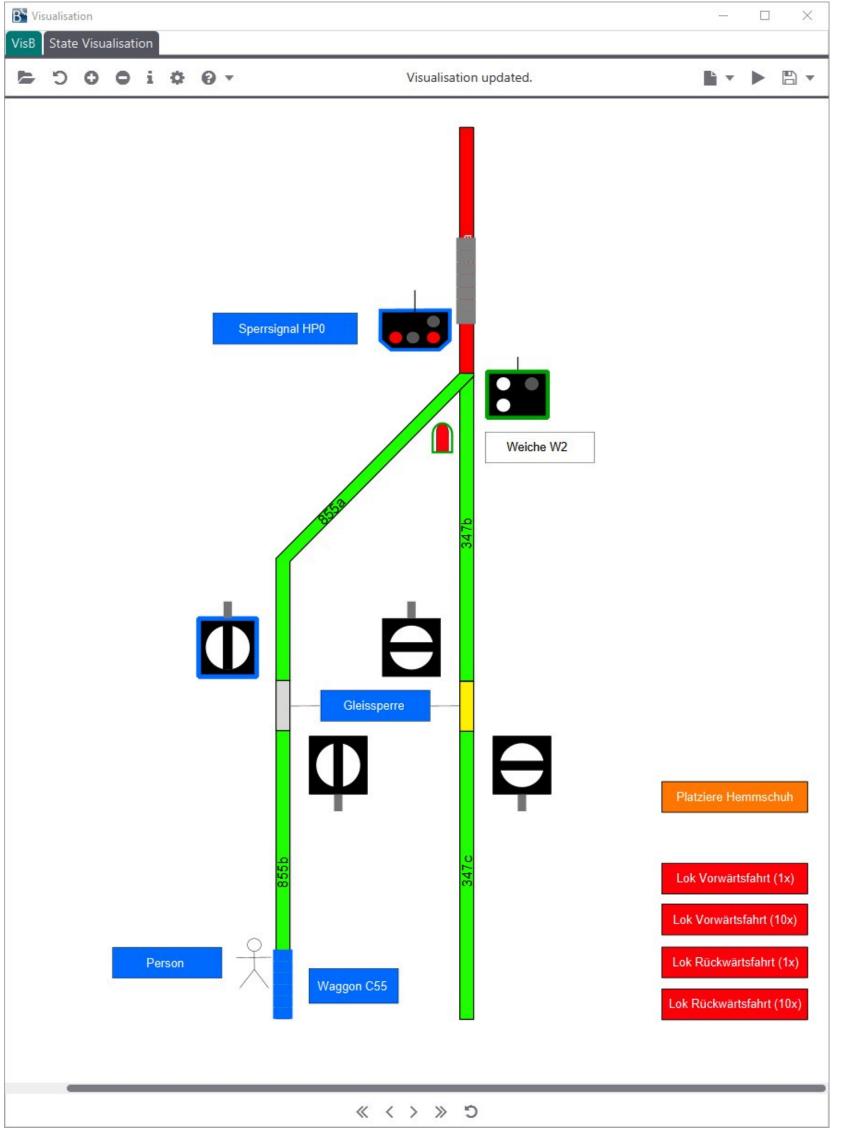
Idea 1: Formal B Model Study impact of AI for behavior of overall system



Formal Model: DEMO



Formal Model



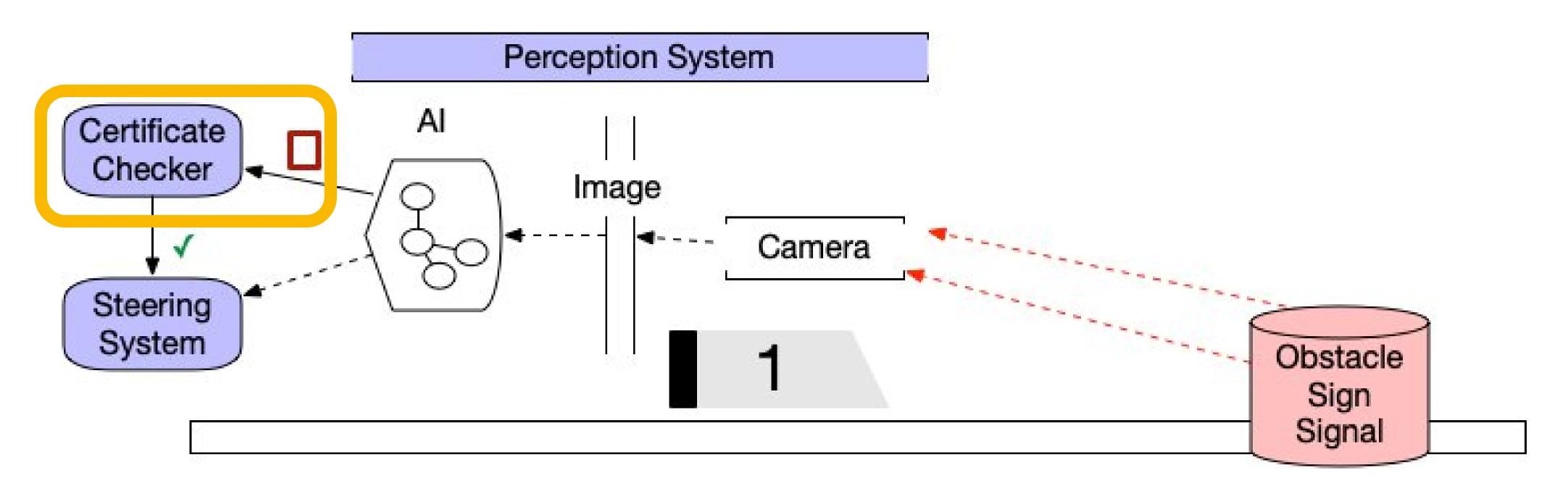
- Requirements (abbreviated):
- Mission Order: Drive along route recognize all objects correctly
- **REC1-5**: Perception system must recognize all objects better than humans
- **SAF1-5**: When all objects are recognized correctly - there are no accidents
 - **PROP1**: Safety-critical situations shall 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

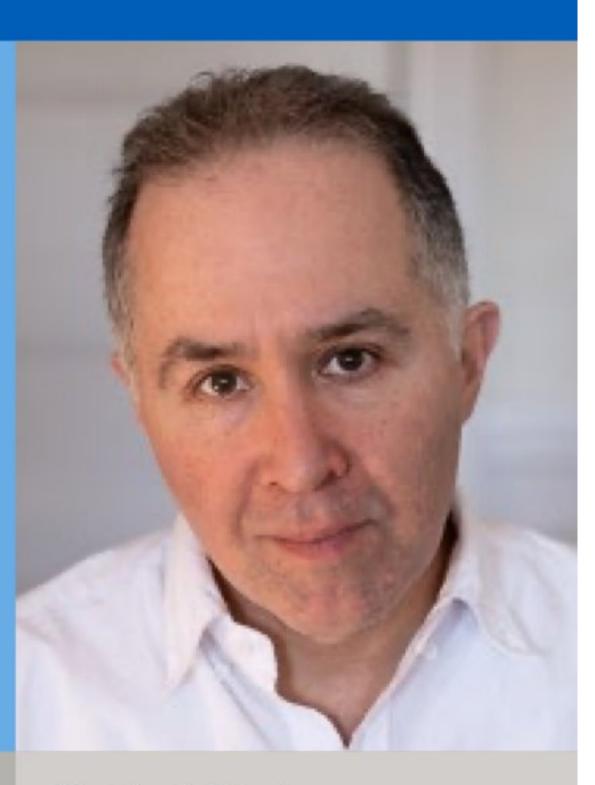
Johns Hopkins Institute for Assured Autonomy and the Department of Computer Science

Present

Certified Control For Autonomous Driving

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

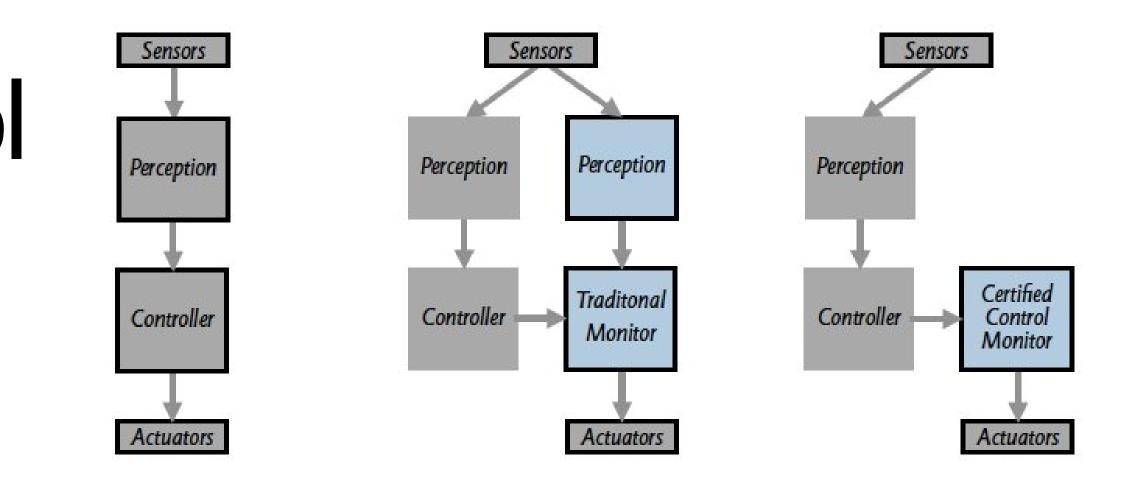
https://iaa.jhu.edu/event/iaa-seminar-series-daniel-jackson-mit/



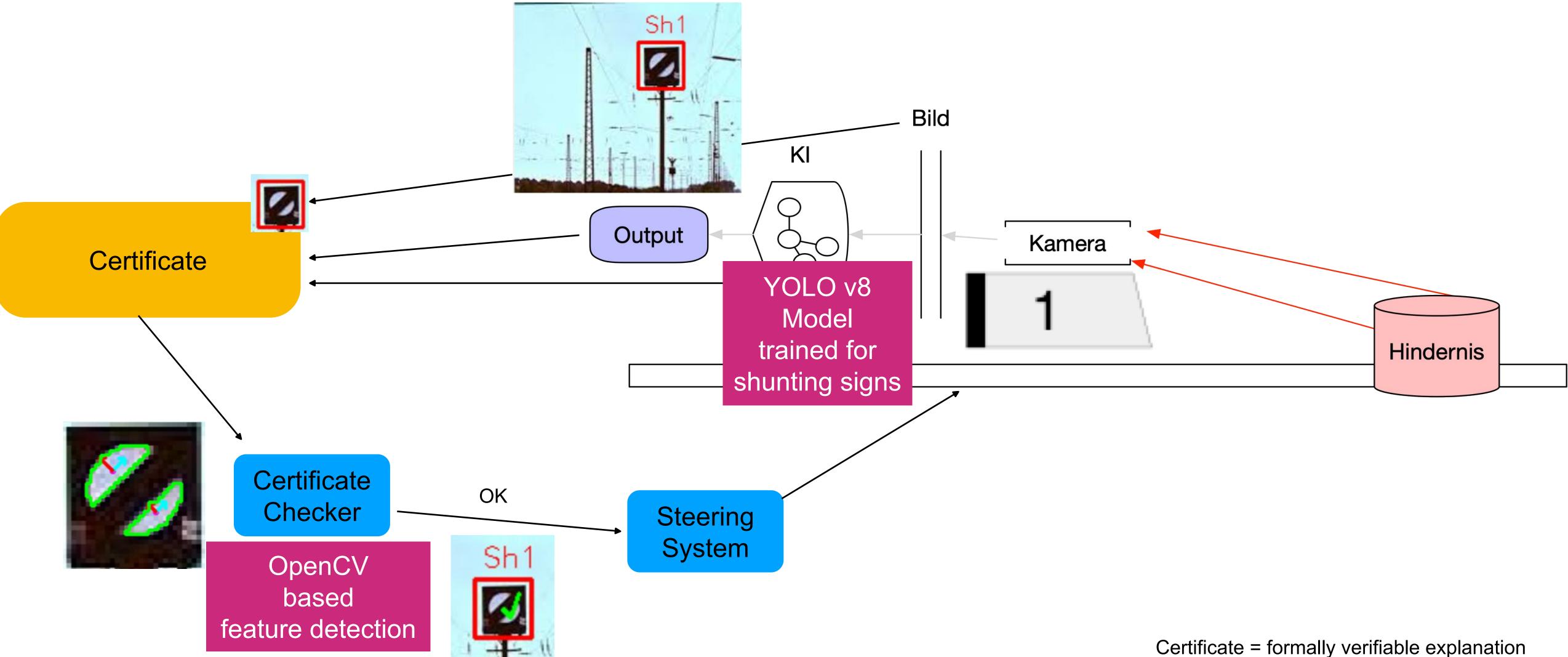
Dr. Daniel Jackson MIT Professor of Computer Science

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

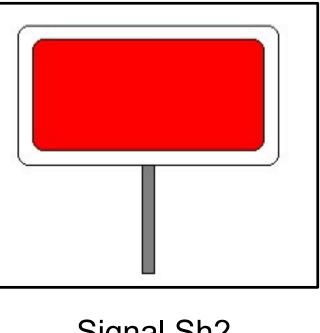


Idea 2: Certified Control Implement and evaluate it in the context of our case study



Erroneous classification detected by checker:





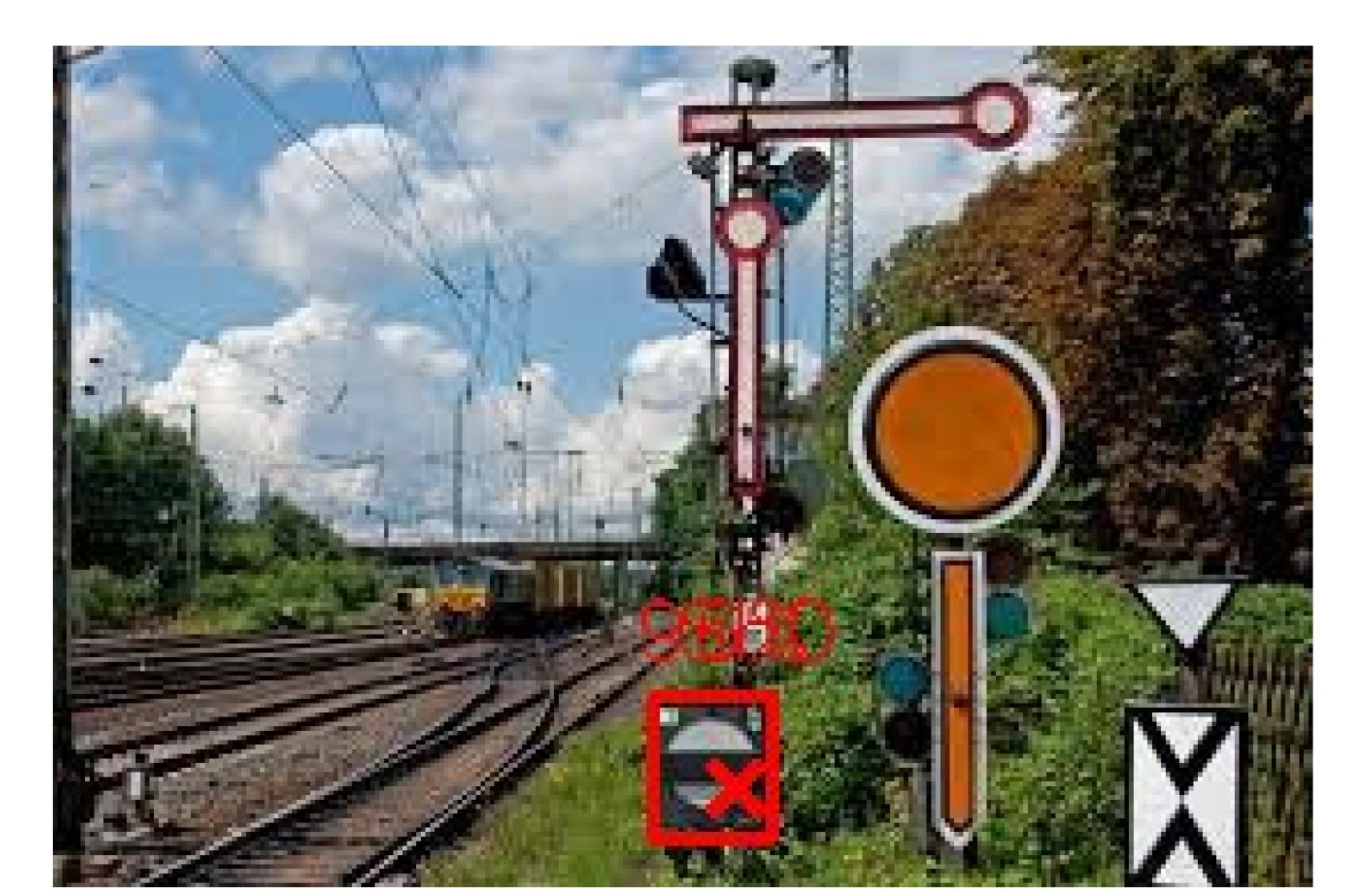
Signal Sh2 Protective STOP



Correct classification rejected by checker:

the certificate checker may also reject correct classifications





Certified Control: Challenges/Problems

- Not a proof of correct classification results
- Individual Solution for each safety property required => expensive
- Does not address non detections (false negatives) ullet

Solutions - if sign is not detected by AI:

expected,

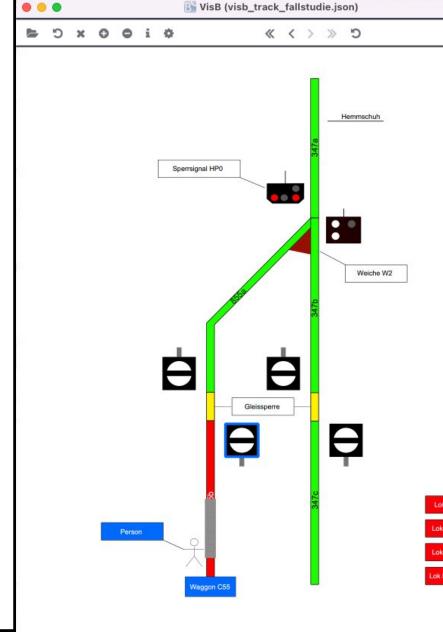
2. or accept AI can make errors - conduct a probabilistic analysis (SimB)

• Gain Accuracy at the cost of Recall (Trade off depends on the exact implementation)

- 1. provide location of signals/signs go into a safe mode when no sign detected where

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.	
Operations	State View State Visualisation Edit	 Statistics (states 53 of 492)
	🖺 🖸 😮 Rangierfah	► Verifications
 MoveLokForwards(frmt=B855b, nxt=B855b, back=B855b, new_froi MoveLokForwards(frmt=B855b, prev=B855a, back=B855b, new_froi MoveLokBackwards(frmt=B855b, prev=B855a, back=B855b, new_froi MoveLokBackwards(frmt=B855b, prev=B855a, back=B855b, new_froi MoveLokBackwards(frmt=B855b, prev=B855a, back=B855b, new_i MoveLokBackwards(frmt=B855b, prev=B855a, back=B855b, new_i MoveLokBackwards(frmt=B855a, B2=B855b) DeactivateDerailer(B1, B2) DeactivateDerailer(B1=B347b, B2=B347c) DeactivateDerailer(B1=B355a, B2=B855b) SwitchSignalToSh0(frmt, nxt) SwitchSignalToSh0(frmt, nxt) SwitchSignalToSh0(frmt, nxt) DetectObject(o=C55, B=B855b) DetectObject(o=C55, B=B855b) DetectObject(o=C55, B=B855b) DetectObject(o=C55, B=B855b) DetectObject(o=C55, B=B655b) DetectObj	<pre>I MACHINE Rangierfahrt@_prob_fallstudie EXTENDS Rangierfahrt@_ DEFINITIONS I VISB_SON_FILE == "visb_track_fallstudie.json"; CUSTOM_GRAPH_DOES == BLOCKS; CUSTOM_GRAPH_EDGES == next PCONSTANTS B347a, B347b, B347c, B855a, B855b, C55, Person PROPERITES OBJECTS = {lok, C55, Person} & Cased({lok, C55, Person} & Cased(</pre>	Project Machines Status Preferences Project Machines Status Preferences Project Rangierfahrt0_prob_fallstudie Rangierfahrt0_prob_fallstudie Rangierfahrt0_prob_fallstudie Rangierfahrt0_prob_fallstudie Rangierfahrt0_prob_fallstudie Rangierfahrt0_prob_fallstudie History (state 54 of 54) Status Status Machines Status Status Position▲ O root 1 SETUP_CONSTANTS 2 2 INITIALISATION 3 3 MoveLokForwards(frnt=B34 4 MoveLokForwards(frnt=B34 5 MoveLokForwards(frnt=B34

can satura da	Teacher ann	WS-W-	20.00	1
L.		6	*	
			1	
States -	Infahat i	4.4		
ok Vorwär k Vorwärt				
k Rückwä				
Rückwär				
RUCKWAI	usianini (TUX		
tatistic	s (sta	tes 5	i3 o	f 492)
erificat	tions			
roject		Ţ	2	le le
chines	Stat	us	Pre	ferences Proje
Rangi ngiertah	ierfal	hrt0_fi	_pr	ob_fallstudie udie.mch

Models are available at:

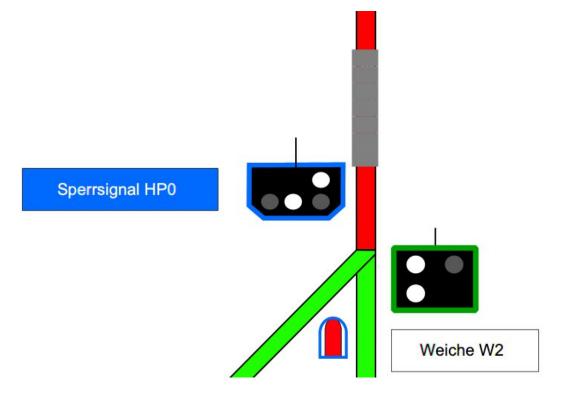
https://github.com/hhu-stups/kilok_shunting_model

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

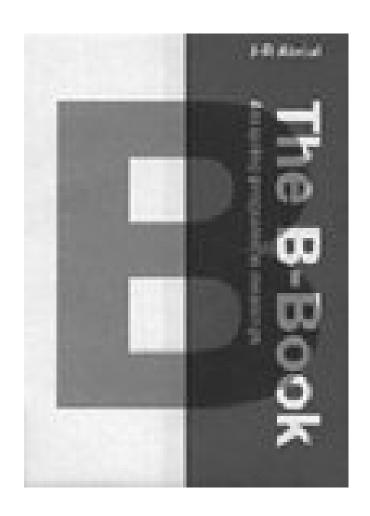


Formal B Model

AI System



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

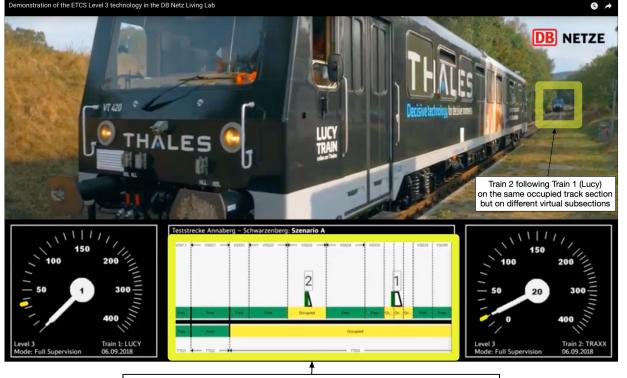


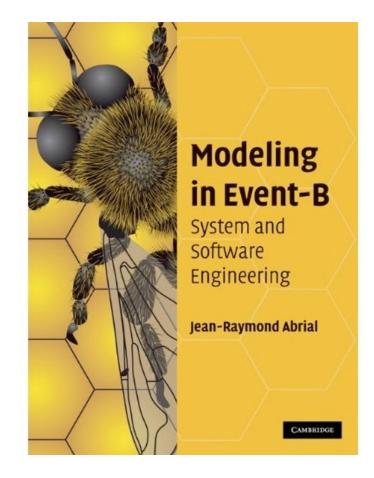


Software (30% of CBTC systems worldwide use B software)



Data & Config. Validation





System Specification & Executable Model

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

System Analysis & Safety Case



	• •		B Rangierfahrt_KI_1_1.mch - Rangierfahrt_Project - ProB
Оре	erationen		Zustandsansicht Bearbeiten
«	< > 2	C «	Image: Second state Image: Second state Imag
►	RF_MoveL	LokBackwards(frnt=B347a, prev=B347a, back=B347a,	new_front=B347a, new_back Name
•	RF_MoveL	LokBackwards(frnt=B347a, prev=B347a, back=B347a,	new_front=B347a, new_back VARIABLES
-	-	LokBackwards(frnt=B347a, prev=B347a, back=B347a,	new_front=B347a, new_back
2		LokBackwards(frnt=B347a, prev=B347a, back=B347a,	new_front=B347a, new_back
5		LokBackwards(frnt=B347a, prev=B347a, back=B347a, LokBackwards(frnt=B347a, prev=B347a, back=B347a,	
		LokBackwards(frnt=B347a, prev=B347a, back=B347a,	
•	RF_MoveL	LokBackwards(frnt=B347a, prev=B347a, back=B347a,	new front=B347a, new back
•		LokBackwards(frnt=B347a, prev=B347a, back=B347a,	new_front=B347a, new_back
2	_	LokBackwards(frnt=B347a, prev=B347a, back=B347a,	
5	-	LokForwards(frnt=B347a, nxt=B855a, back=B347a, ne LokForwards(frnt=B347a, nxt=B855a, back=B347a, ne	
	-	LokForwards(frnt=B347a, nxt=B855a, back=B347a, ne	w front=B347a new back=B
•		LokForwards(frnt=B347a, nxt=B855a, back=B347a, ne	w_front=B347a, new_back=B $\forall (0,b1,b2) \cdot (0 \mapsto b1 \in ENV_occ \land 0 \mapsto b2 \in ENV_occ \land$ true
•	NAME OF TAXABLE PARTY.	LokForwards(frnt=B347a, nxt=B855a, back=B347a, ne	
-	-	LokForwards(frnt=B347a, nxt=B855a, back=B347a, ne	
-		LokForwards(frnt=B347a, nxt=B855a, back=B347a, ne LokForwards(frnt=B347a, nxt=B855a, back=B347a, ne	uu frant=P247a, nau haak=P
5		LokForwards(frnt=B347a, nxt=B855a, back=B347a, ne	w front=B347a, new back=B
		LokForwards(frnt=B347a, nxt=B855a, back=B347a, ne	\mathbb{W} \mathbb{W} \mathbb{W} \mathbb{W} \mathbb{W} \mathbb{W} \mathbb{W} \mathbb{W} \mathbb{W} \mathbb{W} \mathbb{W} \mathbb{W}
۲	ENV_Start	tMovePoint(Block=B347a, N1=B347b, N2=B855a)	
•		MovePoint(Block, N1, N2)	▼ Visualisierung
2		vateDerailer(B1=B347b, B2=B347c) vateDerailer(B1=B855a, B2=B855b)	VisB Zustandsvisualisierung
0		ctivateDerailer(B1, B2)	
•	Contraction and Contraction	eBrakeShoe_Front(B=B347a, pos=91)	🖙 🗇 i 🌣 🖺 🔻 Visualisierung aktualisiert.
•	ENV_Place	eBrakeShoe_Front(B=B347a, pos=92)	
	_	eBrakeShoe_Front(B=B347a, pos=93)	
1	and the second second second	eBrakeShoe_Front(B=B347a, pos=94) eBrakeShoe_Front(B=B347a, pos=95)	Sperrsignal HP0
5		eBrakeShoe_Front(B=B347a, pos=95)	
	and the second second second	eBrakeShoe_Front(B=B347a, pos=97)	
•	ENV_Place	eBrakeShoe_Front(B=B347a, pos=98)	
		eBrakeShoe_Front(B=B347a, pos=99)	Weiche W2
-		eBrakeShoe_Front(B=B347b, pos=0)	
		noveBrakeShoe_Front(B, pos) chSignalToSh0(B1=B347a, B2=B855a)	
•	1.1.1	chSignalToSh1(B1, B2)	
•	VIS_Detec	ctCorrectObject_Front(reason=wagon)	
•	VIS_Detec	ctDisappearedStopReason_Front(reason)	
öal	licherweise	mehr - MAX_OPERATIONS erreicht	
200	Animation		
			Gleissperre
Nac	hspielen S	Symbolisch Testfallgenerierung	
\odot)		
0	Status	Name	Schritte
D	×	[TR1] Mission_Order	40
D	×	[TR10] Mission_Order10	11
D	0	[TR11] Mission_Order11	15
D	0	[TR12] Mission_Order12	12 Person
D	0	[TR13] Mission_Order13	11 Waggon C55
D	0	[TR14] Mission_Order14	12
D	0	[TR15] Mission_Order15	16
D	0	[TR2] Mission_Order2	41
Ø	0	[TR3] Mission_Order3	41 Interaktive Konsole

				 Statistik (Zustände 47 von 2.103)
			8	Überprüfungen
	Í			▼ Projekt
				Maschinen Status Präferenzen Projekt
				$0 \cdot 1 1$
	true			definitions definitions.def
	true			▶ Environment
	true			Environment.mch Vision
	true			Vision.mch
next))	true			Control Control.mch
/_next ⁻¹)) V occ A	true			► Rangierfahrt
V_000 A	true			Rangierfahrt.mch
ok → b1 ∈	true			Rangierfahrt_KI Rangierfahrt_KI.ref
or of chi	true			Rangierfahrt_KI_1_1
SIGNAL				Rangierfahrt_Kl_1_1.mch
SIGNALS				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
	100 00 000 000 000 000 000 000	-		
Visualisier	ung aktualisiert. 🗨 🤤	20	•	
Visualisier	ung aktualisiert. 🗨 G	0	•	
	ung aktualisiert. 🗨 G	0	•	
Visualisier	ung aktualisiert. 🗨 G	20	•	Verlauf (Zustand 13 von 40)
	ung aktualisiert. 🗨 G		•	Verlauf (Zustand 13 von 40) ≪ く > ≫ つ
	ung aktualisiert. \bigcirc C			
	ung aktualisiert. Q C			≪ < > ≫ つ
			P	≪ < > ≫ づ E ▼ ? Position▲ Transition 11 VIS_DetectCorrectSignal_Front(B1=B347a, B2=B855a)
			P 11	
			Р 11 12	
			P 11 12 13 74	
			P 11 12 13 74	
			P 11 12 13 14 15	
			P 11 12 13 14 15 16	Image: Sector
			P 11 12 13 14 15 16 17	
			P 11 12 13 14 15 16 17 18	Image: Sector
			P 11 12 13 14 15 16 17 18 19	
			P 11 12 13 14 15 16 17 18 15 20 21 22	Image: Sector
	Weiche W2		P 11 12 13 14 15 16 17 18 19 20 21 22 23	
			P 111 122 133 144 155 166 177 185 200 211 222 233 244	X X
	Weiche W2		P 111 122 133 144 155 166 177 188 199 200 211 222 233 244 255	
	Weiche W2 Voiche W2 Voiche W2 Voiche W2 Voiche W2 Voiche W2 Voiche W2 Voiche W2 Voiche W2 Voiche W2		P 111 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26	Normalization Image: Content of the second seco
	Weiche W2 Weiche W2		P 111 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27	Note: Note: <td< th=""></td<>
C.S.	Weiche W2 Description Lock Vorwärtsfahrt (1x) Lock Vorwärtsfahrt (10x)		P 111 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28	Position I Transition II VIS_DetectCorrectSignal_Front(B1=B347a, B2=B855a) III VIS_DetectCorrectSignal_Front(B1=B347a, nxt=B855a, back=B347a, new_front=B347a, new III RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B347a, new III RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B347a, new III RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B357a, new III RF_MoveLokForwards(frnt=B347a, nxt=B855b, back=B347a, new_front=B855a, new III RF_MoveLokForwards(frnt=B355a, nxt=B855b, back=B347a, new_front=B855a, new III RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855b, new <td< th=""></td<>
C.S.	Weiche W2 Weiche W2		P 111 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29	Non-
i i i i i i i i i i i i i i i i i i i	Weiche W2 Weiche W2		P 111 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28	Non- Non- Non- Non- 11 VIS_DetectCorrectSignal_Front(B1=B347a, B2=B855a) RF_MoveLokForwards(frmt=B347a, nxt=B855a, back=B347a, new_front=B347a, new 12 RF_MoveLokForwards(frmt=B347a, nxt=B855a, back=B347a, new_front=B347a, new 13 RF_MoveLokForwards(frmt=B347a, nxt=B855a, back=B347a, new_front=B3547a, new 14 RF_MoveLokForwards(frmt=B347a, nxt=B855b, back=B347a, new_front=B855a, new 15 RF_MoveLokForwards(frmt=B355a, nxt=B855b, back=B347a, new_front=B855a, new 16 RF_MoveLokForwards(frmt=B855a, nxt=B855b, back=B347a, new_front=B855a, new 17 RF_MoveLokForwards(frmt=B855a, nxt=B855b, back=B347a, new_front=B855a, new 18 RF_MoveLokForwards(frmt=B855a, nxt=B855b, back=B347a, new_front=B855a, new 19 RF_MoveLokForwards(frmt=B855a, nxt=B855b, back=B347a, new_front=B855a, new 19 RF_MoveLokForwards(frmt=B855a, nxt=B855b, back=B357a, new_front=B855a, new 20 RF_MoveLokForwards(frmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new 21 RF_MoveLokForwards(frmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new 22 RF_MoveLokForwards(frmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new 23 RF_MoveLokForwards(frmt=B855a, nxt=B855b, back=B855a, new_front=B855b, new 24 VIS_DetectC

• •							🕒 Rangierfahrt_KI_1
Oper	ationen						Zustandsansicht Bearbeiten
«	< >	с «	o = 20	-	Q	0	Zustand filtern
•	RF_Move	LokBackwards(frnt=B855a, prev=B347a, back=B347a, ne	w_front=B3	847a,	new_t	back	Name
•	_	LokBackwards(frnt=B855a, prev=B347a, back=B347a, ne	_				VARIABLES
	and the factor of	LokBackwards(frnt=B855a, prev=B347a, back=B347a, ne	the second second second		and the second second	1000	▶ CONSTANTS
	_	LokBackwards(frnt=B855a, prev=B347a, back=B347a, ne LokBackwards(frnt=B855a, prev=B347a, back=B347a, ne	-				▶ SETS
		LokBackwards(frnt=B855a, prev=B347a, back=B347a, ne	and the second se				▼ INVARIANT
		LokBackwards(frnt=B855a, prev=B347a, back=B347a, ne	_				[=] dom(ENV_occ) = ENV_OBJECTS
	RF_Move	LokBackwards(frnt=B855a, prev=B347a, back=B347a, ne	w_front=B3	847a,	new_t	back	
•	RF_Move	LokBackwards(frnt=B855a, prev=B347a, back=B347a, ne	w_front=B3	847a,	new_t	back	[⊆] ENV_next ⊆ ENV_TRK
•		LokBackwards(frnt=B855a, prev=B347a, back=B347a, ne		0.0000000000000000000000000000000000000			[∈] ENV_next ∈ ENV_BLOCKS → ENV_BLOCKS
	_	LokForwards(frnt=B855a, nxt=B855b, back=B347a, new_			_		[finite] closure1(ENV_next) ∈ FIN(closure1(ENV_
-		LokForwards(frnt=B855a, nxt=B855b, back=B347a, new_ LokForwards(frnt=B855a, nxt=B855b, back=B347a, new			-		[finite] closure1(ENV_next ⁻¹) ∈ FIN(closure1(ENV)
-	as a vote to	LokForwards(frnt=B855a, nxt=B855b, back=B347a, new_			_	Contract of the last	▶ $[\forall] \forall (o,b1,b2) \cdot (o \mapsto b1 \in ENV_occ \land o \mapsto b2 \in EN$
	_	LokForwards(frnt=B855a, nxt=B855b, back=B347a, new_		2	_		▶ [⊆] ENV_active_derailers ⊆ ENV_DERAILERS
	North Contraction	LokForwards(frnt=B855a, nxt=B855b, back=B347a, new_			-		▶ $[\forall] \forall (b1,b2) \cdot (b1 \mapsto b2 \in ENV_active_derailers \land I$
•	A CONTRACTOR OF A CONTRACTOR O	LokForwards(frnt=B855a, nxt=B855b, back=B347a, new_			1000		[∈] ENV_brake_shoes ∈ ENV_BLOCKS → Z
-	_	LokForwards(frnt=B855a, nxt=B855b, back=B347a, new_					[∈] ENV_signal_states ∈ ENV_SIGNALS → ENV
	and the second second	LokForwards(frnt=B855a, nxt=B855b, back=B347a, new_			the second	and the second se	[∀] ∀s1·(s1 ∈ ENV SIGNALS ⇒ ∀s2·(s2 ∈ ENV S)
	-	LokForwards(frnt=B855a, nxt=B855b, back=B347a, new_	front=B855	a, nev	w_bac	k=B	
		tMovePoint(Block, N1, N2) MovePoint(Block, N1, N2)					▼ Visualisierung
		vateDerailer(B1=B347b, B2=B347c)					
	_	vateDerailer(B1=B855a, B2=B855b)					VisB Zustandsvisualisierung
•	ENV_Dea	ctivateDerailer(B1, B2)					
•		eBrakeShoe_Front(B=B347a, pos=0)					► ') i ¢ 🖺 •
-		eBrakeShoe_Front(B=B347a, pos=1)				_	
		eBrakeShoe_Front(B=B347a, pos=2) eBrakeShoe_Front(B=B347a, pos=3)					
	and the second se	eBrakeShoe_Front(B=B347a, pos=3)					Sperrsignal HP0
	_	eBrakeShoe Front(B=B347a, pos=5)					
	and the ball of the second state	eBrakeShoe_Front(B=B347a, pos=6)					
•	ENV_Plac	eBrakeShoe_Front(B=B347a, pos=7)					
•	_	eBrakeShoe_Front(B=B347a, pos=8)					
	and the second second	eBrakeShoe_Front(B=B347a, pos=9)				_	
	_	noveBrakeShoe_Front(B, pos) tchSignalToSh0(B1, B2)					
0		tchSignalToSh1(B1, B2)					
		ctCorrectObject_Front(reason=wagon)					
•	VIS_Dete	ctDisappearedStopReason_Front(reason)					
ii alt	henueles						
-		mehr - MAX_OPERATIONS erreicht					
v ,	Animation						G
Nach	spielen	Symbolisch Testfallgenerierung					
\odot				5	*	0	
2	Status	Name			Schri	_	
0	×	[TR1] Mission_Order			40		
0	×	[TR10] Mission_Order10			11		
D D	0	[TR11] Mission_Order11			15		
		development There are					
0	0	[TR12] Mission_Order12			12		Person
\odot	0	[TR13] Mission_Order13			11		Wagge
\checkmark	0	[TR14] Mission_Order14			12		
\checkmark	0	[TR15] Mission_Order15			16		
\supset	0	[TR2] Mission_Order2			41		
\supset	0	[TR3] Mission_Order3			41		Interaktive Konsole
		241				100	

Vision mdh Vy. next 1) true Nok - b1 e true SiGNALS	
Vesualisient Q Q Q Vesualisient Q Q Q Vesualisient Vestation Transition Vesualisient Q Q Q Vesualisient Vestation Transition Vesualisient Q Q Q Vesualisient Posterior Transition Vesualisient Q Q Q Vesualisient Posterior Transition Vesualisient Posterior Transition Vesualisient Q Q Q Vesualisient Posterior Posterior Vesualisient Q Q Q Vesualisient Posterior Posterior Vesualisient Posterior Posterior Vesualisient Q Q Q Vestation Posterior Posterior Vestation Posterior Posterior Posterior Poster	
Norman Norman Norman Norman Norman Norman Norman Norman Norman Norman Norman Norman Norman Norman Norman Norman Norman Norman Norman Norm	
Vertad (Zustand 16 von 40) Vertad (Zustan	
use use use use use use use use use use use use vse	
use use use use use use use use use use use use vse	6
Index Index I	
Pue S Luc S Luc S Luc Main Main <t< td=""><td></td></t<>	
S Fue (met)	
Linext) Pue Wienermain Weinermain Wienermain Wienermain Wienermain Wienermain Uber bit e True True True Vision True </td <td></td>	
Parkt ¹)) True W_next ¹)) True Yue Ranglerfahr, R/L Yue Ranglerfahr, K/L Yue Ranglerfahr, K/L Ranglerfahr, K/L Ranglerfahr, K/L	
Paragiofisht nch Paragiofisht nch Yaue Yue Yaue Yaue </td <td></td>	
Tree Ibk +> b1 e tree /_SIGNAL tree /_SIGNAL siGNALS tree SIGNAL vs.anglerfahrt, KL 1, 2a Ranglerfahrt, KL 1, 2b Ranglerfahrt, KL 1, 2b <t< td=""><td></td></t<>	
Iok + b1 e true y SIGNAL. true y SIGNAL. true SIGNAL.S true SIGNAL.S true Ysualisierung aktualisiert. Q Q Q Q T Visualisierung aktualisiert. Visualisierung aktualisierung aktualitierung aktualisierung a	
✓ SIGNAL true SIGNAL true SIGNAL true Namigerfahrt, Kl. J., 2amch Pangierfahrt, Kl. Random, Topology Pangierfahrt, Kl. Random, Topology, mci Visualisierung aktualisiert. Q. Q. Q. Q. Q. Imagerfahrt, Kl. Random, Topology, mci Visualisierung aktualisiert. Verlauf (Zustand 16 von 40) (K. K. Pandom, Topology, mci Position ▲ Transition 11 Vis_DetectCorrectSignal_Front(B1=B347a, new_front=B34 13 RF_MoveLokForwards(Int=B347a, new_front=B34 14 RF_MoveLokForwards(Int=B35a, nat=B35b, back=B347a, new_front=B35 15 RF_MoveLokForwards(Int=B35a, nat=B35b, back=B347a, new_front=B35 16 RF_MoveLokForwards(Int=B35a, nat=B35b, back=B347a, new_front=B35 17 RF_MoveLokForwards(Int=B35a, nat=B35b, back=B347a, new_front=B35 18 RF_MoveLokForwards(Int=B35a, nat=B35b, back=B35a, new_front=B352 19 RF_MoveLo	
✓_SIGNAL true > Rangierfahrt_Kl_1_2a Rangieddahr, Kl_1_2b Rangieddahr, Kl_1_2b Rangierfahrt_Kl_1_2b Rangierfahrt_Kl_1_2b<	
SIGNALS true Rangierdinkt, KJ, 1_2a.min Rangierdinkt, KJ, 1_2b Rangierdinkt, KJ, 1_2b Rangierdinkt, KJ, 1_2b Rangierdinkt, KJ, 2b Rangierdinkt, KJ, Random, Topology Ran	
Visualisierung aktualisiert. Visualisierung aktualisiert. Visual	
Visualisierung aktualisiert. Q <th< td=""><td></td></th<>	
Visualisierung aktualisiert. Visualisierung aktualisiert. Verlauf (Zustand 16 von 40) Verlauf (Zusta	
Verlauf (Zustand 16 von 40) Weiche WZ Verlauf (Zustand 16 von 40)	
Weiche W2 11 VIS_DetectCorrectSignal_Front(B1=B347a, B2=B855a) 12 RF_MoveLokForwards(fmt=B347a, nxt=B855a, back=B347a, new_front=B34 13 RF_MoveLokForwards(fmt=B347a, nxt=B855a, back=B347a, new_front=B34 14 RF_MoveLokForwards(fmt=B347a, nxt=B855a, back=B347a, new_front=B34 15 RF_MoveLokForwards(fmt=B355a, nxt=B855b, back=B347a, new_front=B35 16 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B347a, new_front=B35 18 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B35a, new_front=B35 19 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B35 20 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B35 21 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B35 22 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B35 23 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B35 23 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B35 23 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B35 24 VIS_DetectCorrectSignal_Front(B1=B855a, nxt=B855b, back=B855a, new_front=B35 24 VIS_DetectCorrectSignal_Front(B1=B855a, nxt=B855b, back=B855a, new_front=B35	- 6
Image: Section of the section of th	
Image: Participation of the second secon	
 RF_MoveLokForwards(fmt=B347a, nxt=B855a, back=B347a, new_front=B85 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B85 	
Image: Construction of the sequence	
Image: Platziere Hemmschuh Platziere Hemmschuh Platziere Hemmschuh Image: Platziere He	
Image: Description of the system Image: Description of the system Image: Description of the	
Resepore 18 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855) 19 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855) 20 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855) 21 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855) 22 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855) 23 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855) 24 VIS_DetectCorrectSignal_Front(B1=B855a, B2=B855b) 25 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855)	
Newsperre 19 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855b, back=B855b, back=B855a, new_front=B855b, back=B855b, back=B855b, back=B855a, new_front=B855b, back=B855b, back=B855a, new_front=B855b, back=B855b, back=B855a, new	B855a,
Platziere Hemmischuh 21 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new_front=B855a, nxt=B855b, back=B855a, new_front=B855a, new_front=B855a	B855a, 55a, new
Platziere Hemmschuh 22 RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a, new_	:B855a, 55a, new 55a, 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) 24 VIS_DetectCorrectSignal_Front(B1=B855a, B2=B855b) 25 RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a)	B855a, 55a, new 55a, new 55a, new
Lok Vorwärtsfahrt (1x) 24 VIS_DetectCorrectSignal_Front(B1=B855a, B2=B855b) Lok Vorwärtsfahrt (10x) 25 RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B855a)	B855a, 55a, new 55a, new 55a, new 55a, new
Lok Vorwärtsfahrt (10x) 25 RF_MoveLokForwards(frnt=B855a, nxt=B855b, back=B855a, new_front=B85	8855a, 55a, new 55a, new 55a, new 55a, new 55a, new
Lok volwartsiant (Tox)	8855a, 55a, new 55a, new 55a, new 55a, new 55a, new 55a, new
DE DE MauelakEanunda (Ante DOEEL auto DOEEL Lastin DOEEL Lastin DOEEL Lastin DOEEL	8855a, 55a, new 55a, new 55a, new 55a, new 55a, new 55a, new
26 RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855a, new_front=B855b, back=B855a, new_front=B855b, back=B855a, new_front=B855b, back=B855a, new_front=B855b, back=B855a, new_front=B855b, back=B855b, back=B85	8855a, 55a, new 55a, new 55a, new 55a, new 55a, new 55a, new
on C55 27 RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855a, new_front=B85	B855a, 55a, new 55a, new 55a, new 55a, new 55a, new 55a, new 55a, new 55b, new
Lok Rückwärtsfahrt (10x) 28 RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855a, new_front=B85	B855a, 55a, new 55a, new 55a, new 55a, new 55a, new 55a, new 55b, new 55b, new 55b, new
29 RF_MoveLokForwards(frnt=B855b, nxt=B855b, back=B855b, new_front=B855b, nxt=B855b, back=B855b, new_front=B855b,	B855a, 55a, new 55a, new 55a, new 55a, new 55a, new 55a, new 55b, new 55b, new 55b, new
31 RF_MoveLokForwards(Int=B855b, nxt=B855b, back=B855b, new_front=B855b, n	B855a, 55a, new 55a, new 55a, new 55a, new 55a, new 55a, new 55b, new 55b, new 55b, new 55b, new 55b, new

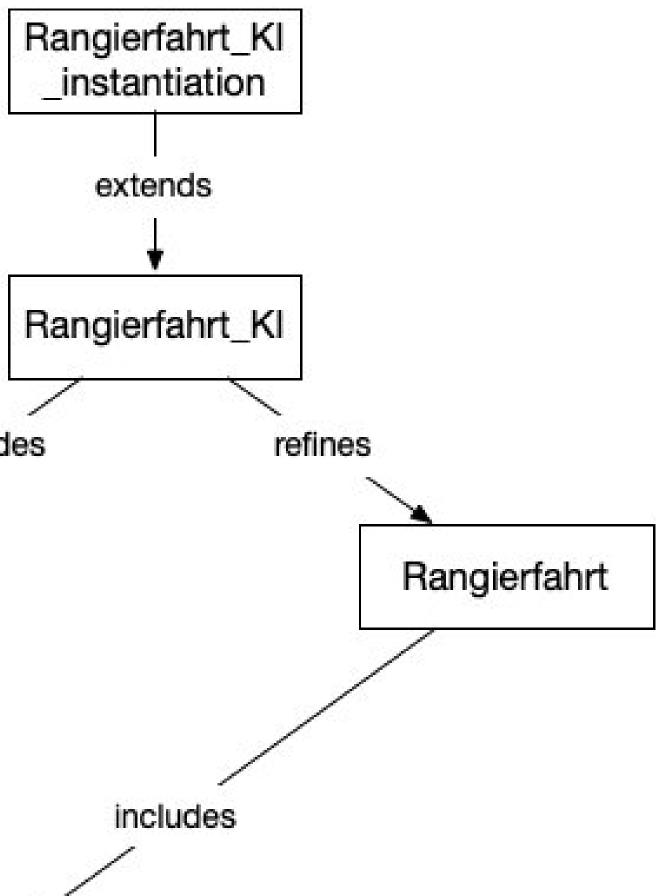
	•			Rangierfahrt_KI_1_1.mch - Rangierfahrt_Project - Propriet - Pro
er	rationen		Z	ustandsansicht Bearbeiten
<	$\langle \rangle$	C «	* · × · Q 0	Zustand filtern
-		LokBackwards(frnt=B855a, prev=B347a, back=B855a, r		Name
	and the second second	LokBackwards(frnt=B855a, prev=B347a, back=B855a, r	ow front=P247a now book	VARIABLES
	and the second se	LokBackwards(frnt=B855a, prev=B347a, back=B855a, r	ew_front=B347a, new_back	CONSTANTS
	_	LokBackwards(frnt=B855a, prev=B347a, back=B855a, i	ew_front=B347a, new_back	SETS
	and the second second	LokBackwards(frnt=B855a, prev=B347a, back=B855a, ı LokBackwards(frnt=B855a, prev=B347a, back=B855a, ı	_none_both a, non_back	INVARIANT true
		LokBackwards(frnt=B855a, prev=B347a, back=B855a, r		
	and the second	LokBackwards(frnt=B855a, prev=B347a, back=B855a, r		
	_	LokBackwards(frnt=B855a, prev=B347a, back=B855a, r		[⊆] ENV_next ⊆ ENV_TRK true
		LokBackwards(frnt=B855a, prev=B347a, back=B855a, r		▶ [∈] ENV_next ∈ ENV_BLOCKS + ENV_BLOCKS true b [€] ENV_next ∈ ENV_BLOCKS + ENV_BLOCKS true
		LokForwards(frnt=B855a, nxt=B855b, back=B855a, nev LokForwards(frnt=B855a, nxt=B855b, back=B855a, nev		[finite] closure1(ENV_next) ∈ FIN(closure1(ENV_next)) true (finite] closure1(ENV_next1) ∈ FIN(closure1(ENV_next1))
		LokForwards(frnt=B855a, nxt=B855b, back=B855a, nev		▶ [finite] closure1(ENV_next ⁻¹) \in FIN(closure1(ENV_next ⁻¹)) true ▶ $\mathbb{N}^{1} \mathbb{N}^{2}$ true
	the second second second	LokForwards(frnt=B855a, nxt=B855b, back=B855a, nev		▶ $[\forall] \forall (o,b1,b2) \cdot (o \mapsto b1 \in ENV_occ \land o \mapsto b2 \in ENV_occ \land$ true
	A CONTRACTOR OF A	LokForwards(frnt=B855a, nxt=B855b, back=B855a, nev		► [⊆] ENV_active_derailers ⊆ ENV_DERAILERS true ► [V] V(b1 b2) (b1 v b2 c ENV_active_derailers + lok v b1 c true
	_	LokForwards(frnt=B855a, nxt=B855b, back=B855a, nev LokForwards(frnt=B855a, nxt=B855b, back=B855a, nev		▶ $[\forall] \forall (b1,b2) \cdot (b1 \mapsto b2 \in ENV_active_derailers \land lok \mapsto b1 \in$ true
	A CONTRACTOR OF A CONTRACTOR OFTA CONTRACTOR O	LokForwards(Int=B855a, nxt=B855b, back=B855a, nev LokForwards(frnt=B855a, nxt=B855b, back=B855a, nev		[∈] ENV_brake_shoes ∈ ENV_BLOCKS + Z true true
	_	LokForwards(frnt=B855a, nxt=B855b, back=B855a, nev		[∈] ENV_signal_states ∈ ENV_SIGNALS → ENV_SIGNAL true
	_	LokForwards(frnt=B855a, nxt=B855b, back=B855a, nev	_front=B855a, new_back=B	▶ $[\forall] \forall s1 \cdot (s1 \in ENV_SIGNALS \Rightarrow \forall s2 \cdot (s2 \in ENV_SIGNALS true$
	- Sector - Constants	tMovePoint(Block=B347a, N1=B347b, N2=B855a)		
		MovePoint(Block, N1, N2) vateDerailer(B1=B347b, B2=B347c)		Visualisierung
		vateDerailer(B1=B855a, B2=B855b)		isB Zustandsvisualisierung
		ctivateDerailer(B1, B2)		🗲 🔿 i 🏟 🖺 🔻 Visualisierung aktualisiert.
		eBrakeShoe_Front(B=B347a, pos=0)		b 3 i 🌣 🖹 🔻 Visualisierung aktualisiert.
		eBrakeShoe_Front(B=B347a, pos=1) eBrakeShoe Front(B=B347a, pos=2)		
		eBrakeShoe_Front(B=B347a, pos=3)		Sperrsignal HP0
		eBrakeShoe_Front(B=B347a, pos=4)		
	A CONTRACTOR	eBrakeShoe_Front(B=B347a, pos=5)		
	_	eBrakeShoe_Front(B=B347a, pos=6) eBrakeShoe_Front(B=B347a, pos=7)		
	and the second states of	eBrakeShoe_Front(B=B347a, pos=8)		Weiche W2
		eBrakeShoe_Front(B=B347a, pos=9)		
	_	noveBrakeShoe_Front(B, pos)		
	and the second second	tchSignalToSh0(B1, B2) tchSignalToSh1(B1, B2)		
	_	ctCorrectObject_Front(reason=wagon)		
	and the second second	ctDisappearedStopReason_Front(reason)		
1	cherwoise	mehr - MAX OPERATIONS erreicht		
	Animation			
				Gleissperre
h	nspielen	Symbolisch Testfallgenerierung		
			⊨ • Ø	
	Status	Name	Schritte	
	×	[TR1] Mission_Order	40	6
	×	[TR10] Mission_Order10	11	
	0	[TR11] Mission_Order11	15	
	0	[TR12] Mission_Order12	12	Person
	0	[TR13] Mission_Order13	11	Waggon C55
	0	[TR14] Mission_Order14	12	
	0	[TR15] Mission_Order15	16	
	0	[TR2] Mission_Order2	41	
		[TR3] Mission Order3	44	
	0	[TK3] MISSION_OIDEIS	41	Interaktive Konsole

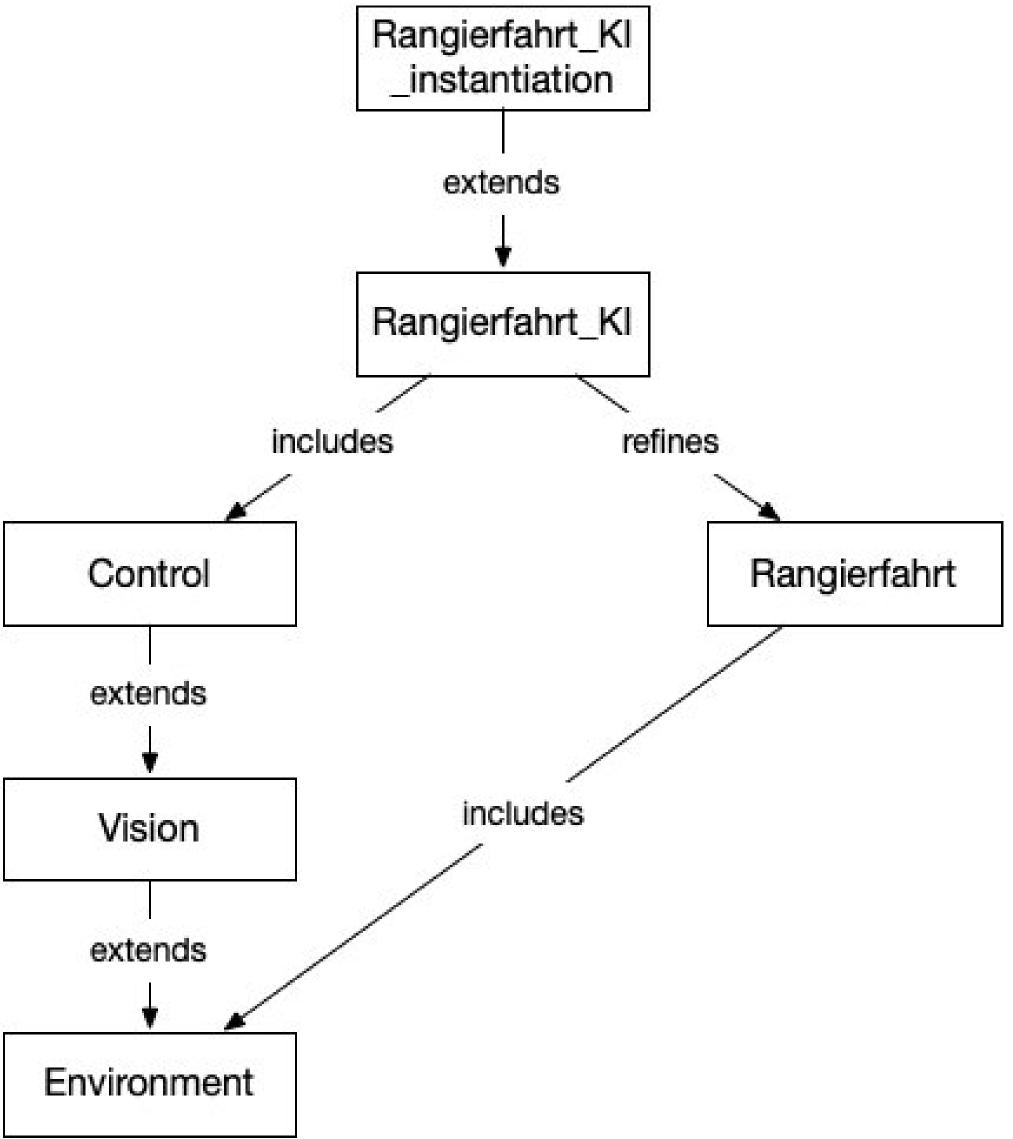
				Statistik (Zustände 47 von 2.103)
			0	
	1			
			W	We Projekt
			_	Maschinen Status Präferenzen Projekt
				$\mathbf{O} \cdot \mathbf{v} \mathbf{O}$
	true			▶ definitions
	true			definitions.def Environment
	true			Environment.mch
KS	true			Vision.mch
V_next))	true			▶ Control
NV_next ⁻¹))	true			Control.mch
ENV_occ A	true			Rangierfahrt Rangierfahrt.mch
8	true			▶ Rangierfahrt_KI
∧ lok → b1 ∈				Rangierfahrt_KI.ref
VV SIGNAL	true			Rangierfahrt_Kl_1_1.mch
/ SIGNALS				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
			0	
Visualisie	rung aktualisiert.	Ψe		
				Verlauf (Zustand 22 von 40)
				Verlauf (Zustand 22 von 40)
				Verlauf (Zustand 22 von 40) ≪ く > » つ P マ ?
	Weiche W2			
	Weiche W2			≪ < > ≫ つ
	Weiche W2			≪ < > ≫ づ E ▼ ? Position▲ Transition 11 VIS_DetectCorrectSignal_Front(B1=B347a, B2=B855a)
	Weiche W2			≪ < > ≫ づ E ▼ ② Position▲ Transition 11 VIS_DetectCorrectSignal_Front(B1=B347a, B2=B855a) 12 RF_MoveLokForwards(frnt=B347a, nxt=B855a, back=B347a, new_front=B347a, new.
	Weiche W2			Image: Constraint of the system Image: Constraint of the system Image: Constraint of the system Image: Constraint of the system Image: Constraint of t
	Weiche W2			Image: Constraint of the system
	Weiche W2			Image: Sector
Gleissperre	Weiche W2			Constition Image: Constition 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=B347a, 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=B355a, new_front=B855a, new.
đ	Weiche W2			
đ	Weiche W2			
đ	Veiche W2	Platziere Hemmschuh		
đ	Veiche W2	Platziere Hemmschuh		
đ	Veiche W2	Platziere Hemmschuh		
đ				Image: Section A Transition 11 VIS_DetectCorrectSignal_Front(B1=B347a, B2=B855a) 12 RF_MoveLokForwards(frmt=B347a, nxt=B855a, back=B347a, new_front=B347a, new. 13 RF_MoveLokForwards(frmt=B347a, nxt=B855a, back=B347a, new_front=B347a, new. 14 RF_MoveLokForwards(frmt=B347a, nxt=B855a, back=B347a, new_front=B855a, new. 15 RF_MoveLokForwards(frmt=B35a, nxt=B855b, back=B347a, new_front=B855a, new. 16 RF_MoveLokForwards(frmt=B855a, nxt=B855b, back=B347a, new_front=B855a, new. 17 RF_MoveLokForwards(frmt=B855a, nxt=B855b, back=B347a, new_front=B855a, new. 18 RF_MoveLokForwards(frmt=B855a, nxt=B855b, back=B347a, new_front=B855a, new. 19 RF_MoveLokForwards(frmt=B855a, nxt=B855b, back=B355a, new_front=B855a, new. 20 RF_MoveLokForwards(frmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new. 21 RF_MoveLokForwards(frmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new. 22 RF_MoveLokForwards(frmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new. 23 <i>RF_MoveLokForwards(frmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new.</i>
đ		Lok Vorwärtsfahrt (1x) Lok Vorwärtsfahrt (10x)		 × × > × 0 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=B35a, 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, back=B855a, new_front=B855b, new.
Gleissperre		Lok Vorwärtsfahrt (1x) Lok Vorwärtsfahrt (10x) Lok Rückwärtsfahrt (1x)		
đ		Lok Vorwärtsfahrt (1x) Lok Vorwärtsfahrt (10x)		NoreLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new. 10 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new. 11 VIS_DetectCorrectSignal_Front(B1=B347a, B2=B855b, back=B347a, new_front=B347a, new. 12 RF_MoveLokForwards(fmt=B347a, nxt=B855a, back=B347a, new_front=B347a, new. 13 RF_MoveLokForwards(fmt=B347a, nxt=B855a, back=B347a, new_front=B857a, new. 14 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B347a, new_front=B855a, new. 15 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B347a, new_front=B855a, new. 16 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B347a, new_front=B855a, new. 17 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B347a, new_front=B855a, new. 18 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B355a, new_front=B855a, new. 20 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new. 21 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new. 23 <i>RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new.</i> 24 VIS_DetectCorrectSignal_Front(B1=B855a, nxt=B855b, back=B855a, new_front=B855b, new. 25 <i>RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855b, new.</i> 26 <i>RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855b, new.</i>
Gleissperre		Lok Vorwärtsfahrt (1x) Lok Vorwärtsfahrt (10x) Lok Rückwärtsfahrt (1x)		NoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new. 10 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new. 11 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B347a, new_front=B847a, new. 13 RF_MoveLokForwards(fmt=B347a, nxt=B855a, back=B347a, new_front=B847a, new. 14 RF_MoveLokForwards(fmt=B347a, nxt=B855a, back=B347a, new_front=B855a, new. 15 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B347a, new_front=B855a, new. 16 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B347a, new_front=B855a, new. 17 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B347a, new_front=B855a, new. 18 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B355a, new_front=B855a, new. 20 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new. 21 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new. 23 <i>RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new.</i> 24 <i>VIS_DetectCorrectSignal_Front(B1=B855a, nxt=B855b, back=B855a, new_front=B855b, new.</i> 25 <i>RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855b, new.</i> 26 <i>RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855b, new.</i> 27 <i>RF_MoveLokForwards(fmt=B855b, nxt=B855b, back=B855a, new_front=B855b, new.</i>
Gleissperre		Lok Vorwärtsfahrt (1x) Lok Vorwärtsfahrt (10x) Lok Rückwärtsfahrt (1x)		NoreLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new. 10 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new. 11 VIS_DetectCorrectSignal_Front(B1=B347a, B2=B855b, back=B347a, new_front=B347a, new. 12 RF_MoveLokForwards(fmt=B347a, nxt=B855a, back=B347a, new_front=B347a, new. 13 RF_MoveLokForwards(fmt=B347a, nxt=B855a, back=B347a, new_front=B857a, new. 14 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B347a, new_front=B855a, new. 15 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B347a, new_front=B855a, new. 16 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B347a, new_front=B855a, new. 17 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B347a, new_front=B855a, new. 18 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B355a, new_front=B855a, new. 20 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new. 21 RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new. 23 <i>RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855a, new.</i> 24 VIS_DetectCorrectSignal_Front(B1=B855a, nxt=B855b, back=B855a, new_front=B855b, new. 25 <i>RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855b, new.</i> 26 <i>RF_MoveLokForwards(fmt=B855a, nxt=B855b, back=B855a, new_front=B855b, new.</i>

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: ullet
 - Neither wagon nor person recognised correctly leads to collision with both Person recognised correctly, but not wagon — leads to collision with wagon \bullet • 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"})

of Points, WP_DT = Update detected track

		<u> </u>				
Model	Operations	Variables/	States	Transitions	Time	Memory
wiodei	Operations	Constants	States	Transmons	(min)	(GB)
CD	13	34	269153	2240046	6.8	1.3
+ WS	14	34	480409	5403158	12.3	2.6
+ WP	15	34	807 001	10733462	23.4	4.8
$+ WP_DT$	15	34	$> \!\! 16785959$	$>\!\!185250252$	>530	$>\!\!80$
complete	22	46	n/a	n/a	n/a	n/a

Model Checking

- CD = Correct Detection, WS = Wrong Signals, WP = Wrong Detection
 - Table 2: Model Checking Results for Selected Reduced Models



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

	D: /	0.10	11.00	01.00	01.40	41 50	F1 00	01 70	71 00	01.00	01 100	101 110
Signal	Distance		11-20	21-30						81-90		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%
	Ι	0.09%	0.09%	31.59%	45.09%	54.09%	63.09%	72.09%	76.59%	81.09%	85.59%	100.00%
Point	Distance	0-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80	81-90	91-100	101-110
Positioning	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%
	Ι	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%
	Ι	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	-	-	2 -	-	-	-	-	-	-	-	-
	Ι	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" v "train reaches the end of 347c" v "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

as good as humans

Validation by Simulation:

SIM(*ending*: "train reaches the end of 855b" v "train reaches the end of 347c" v "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)

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

Validation and Verification: Challenges/Problems

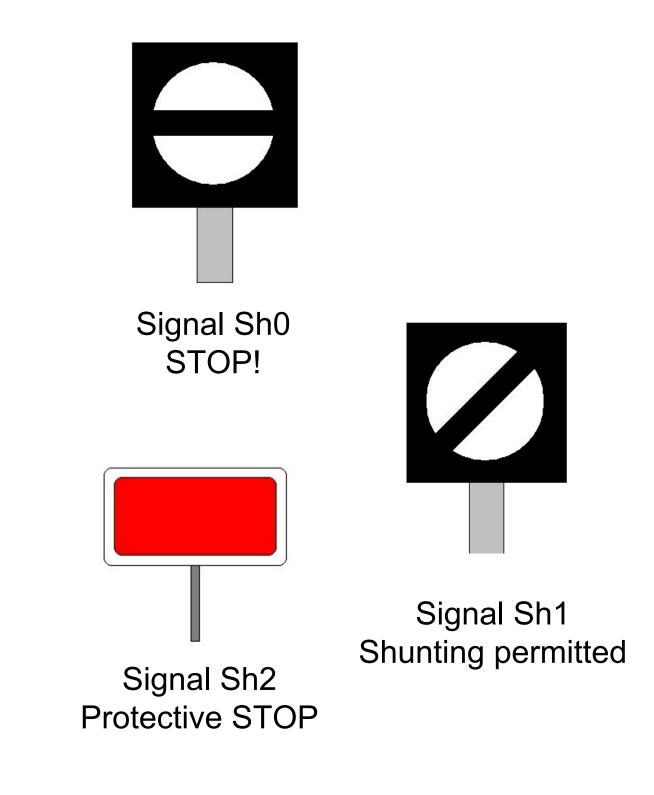
- Validation:
 - Traces only cover parts of state space
 - Probabilities for simulation
- Verification:
 - Model checking struggles with state space explosion
 - Proving very hard on our model
 - certificate checking

Perception System (REC1-REC5) - hard to verify with formal methods - use

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

shunting signs of detected sign



etc.

Related Work

- New standardisation approaches, e.g., UL4600 standard
- Several approaches for verifying neural networks do not scale for our properties and network
- do not cover AI aspect

- Several railway systems: Abrial's interlocking model, CBTC, Hybrid Level 3, ... - especially older railway systems -