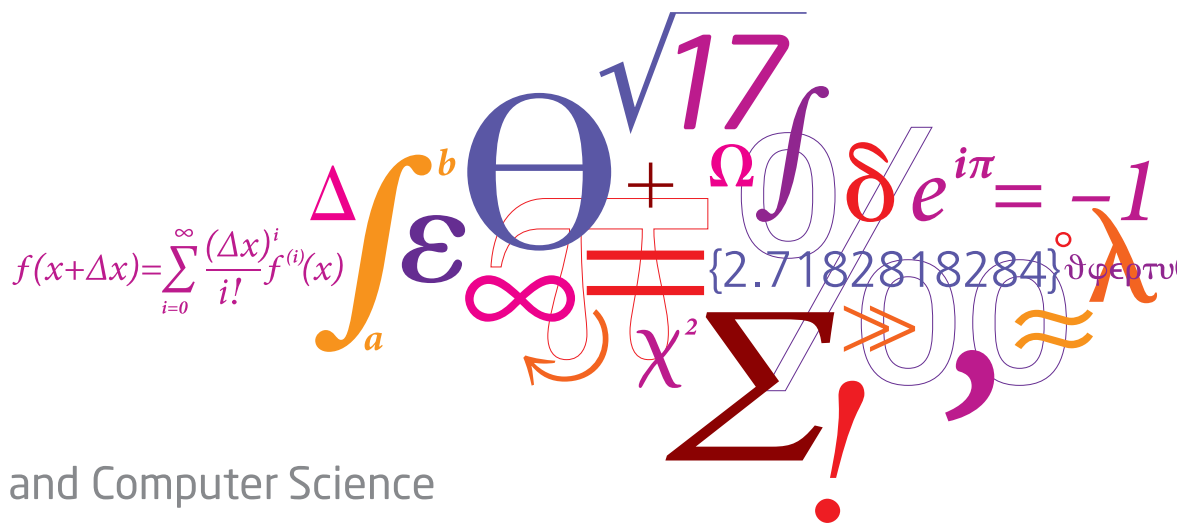
Formal Modeling and Verification of Interlocking Systems Featuring Sequential Release

Linh H. Vu* (lvho@dtu.dk)
Anne E. Haxthausen* (aeha@dtu.dk)
Jan Peleska** (jp@informatik.uni-bremen.de)

* Technical Univesity of Denmark, Denmark
** University of Bremen, Germany



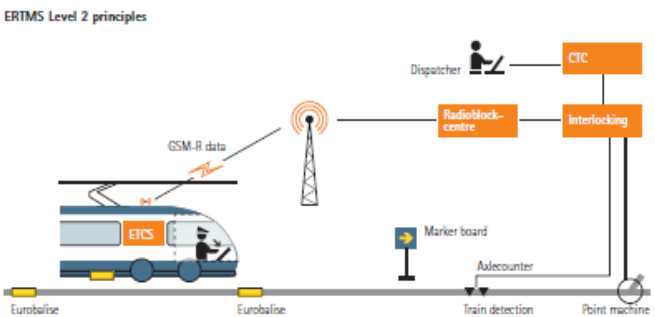
DTU Compute
Department of Applied Mathematics and Computer Science

Background

- **Context**: The Danish Signalling Programme[1] (2009-2021) - replace the railway signalling systems in the entire country with standardized ERTMS/ETCS Level 2
- **ERTMS/ETCS**: European standardized railway traffic management/train control systems → seamless railway travel through Europe
- **RobustRailS**: (Robustness in Railway OperationS[2])
  - Funded by the Danish Strategic Research Council
  - Accompanies the Danish Signalling Programme on a scientific level
- **(One of the) goals**: Provide methods and tools supporting *efficient* modeling and verification of railway control systems (WP.4.1)

→ *How we did that for a case study…*

RobustRailS



Source: ertms.net

[1] http://www.bane.dk/signalprogrammet

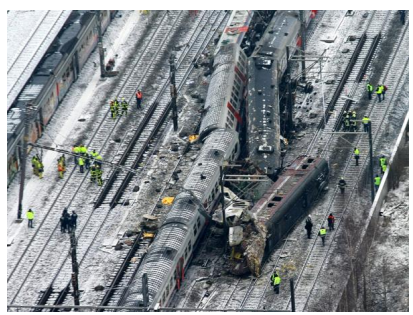
[2] http://robustrails.man.dtu.dk

Agenda

1. Background
2. Interlocking Case Study
3. Toolchain
4. Model and Safety Properties
5. Verification Technique
6. Conclusion

Interlocking Case Study

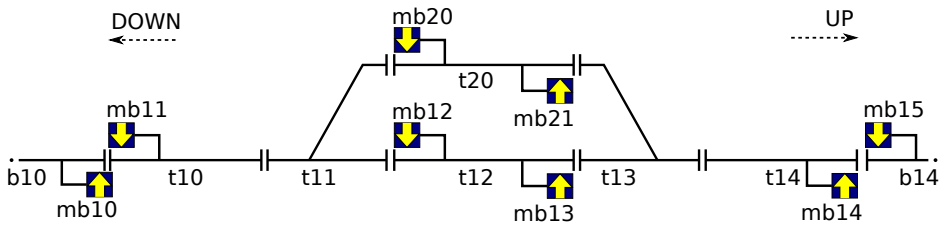
- **Interlocking system**: A component of the signalling system that guides trains *safely* through the (fraction of) railway network under its control
- **Safety-critical**: A vital component with highest safety integrity level (SIL4)
- **Our goal**: Verify high-level safety properties (no collisions, no derailments) for the new Danish interlocking systems
- **Our approach**:
  - Uses *formal methods* (FM) - strongly recommended by CENELEC 50128 standard
  - Defines and uses a *domain-specific language* (DSL) to encapsulate FM
  - Provides tools for *automated* verification

Source: wikipedia.org, skynet.be

Route-based Interlocking Systems

- Reserve a fraction of the network - a *route* - for a train at a time
- *Specification* of a route-based interlocking system consists of
  1. A railway network layout under control

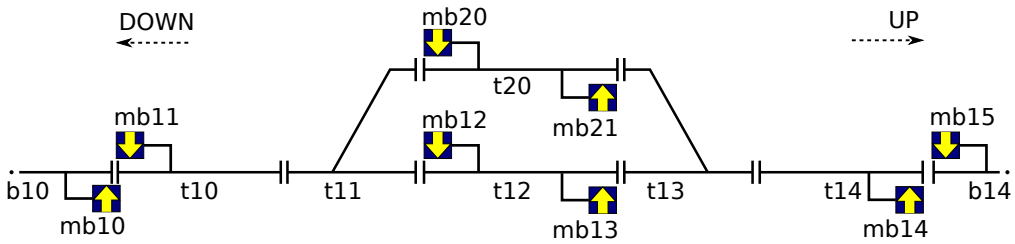


  2. A corresponding interlocking table

ID	src	dest	points	signals	path	conflicts
1	mb10	mb13	t11:p;t13:m	mb11;mb12;mb20	t10;t11;t12	2;3;4;5;7
2	mb10	mb21	t11:m;t13:p	mb11;mb12;mb20	t10;t11;t20	1;3;6;7;8
3	mb12	mb11	t11:p	mb10;mb20	t11;t10	1;2;5;7
4	mb13	mb14	t13:p	mb15;mb21	t13;t14	1;5;6;8
5	mb15	mb12	t11:m;t13:p	mb13;mb14;mb21	t14;t13;t12	1;3;4;6;8
6	mb15	mb20	t11:p;t13:m	mb13;mb14;mb21	t14;t13;t20	2;4;5;7;8
7	mb20	mb11	t11:m	mb10;mb12	t11;t10	1;2;3;6
8	mb21	mb14	t13:m	mb13;mb15	t13;t14	2;4;5;6

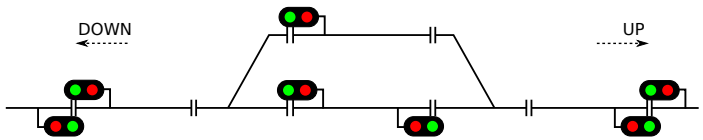
Railway Network Layout

- Geographical arrangment of track-side elements
  - linear sections (t12)
  - points (t11): PLUS (straight) or MINUS (siding) positions
  - marker boards (mb12)

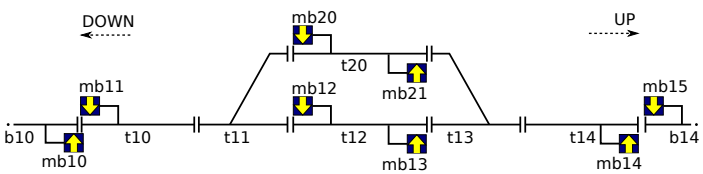


Virtual Signal Concept

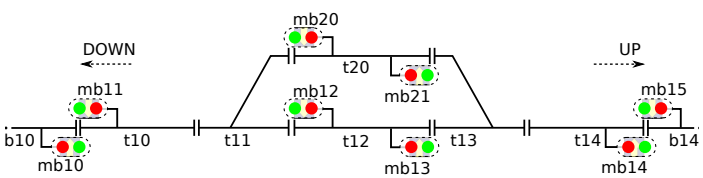
- **ETCS Level 2**: No physical signals on the tracks; instead movement authorities are communicated via on-board computers
  → modeling concept of *virtual signals* associated with marker boards



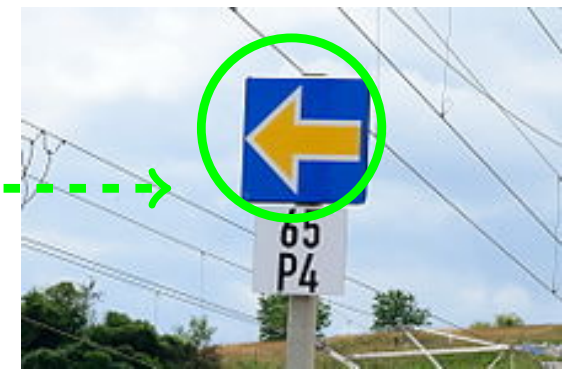
(a) Physical signals



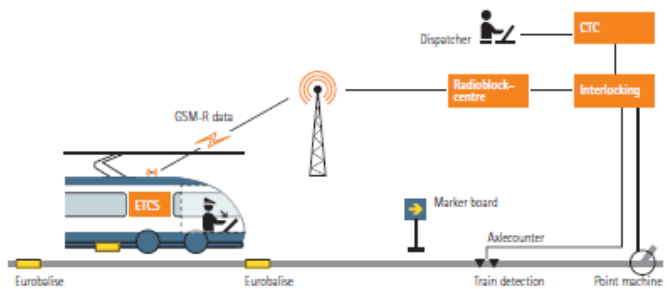
(b) Marker boards



(c) Virtual Signals



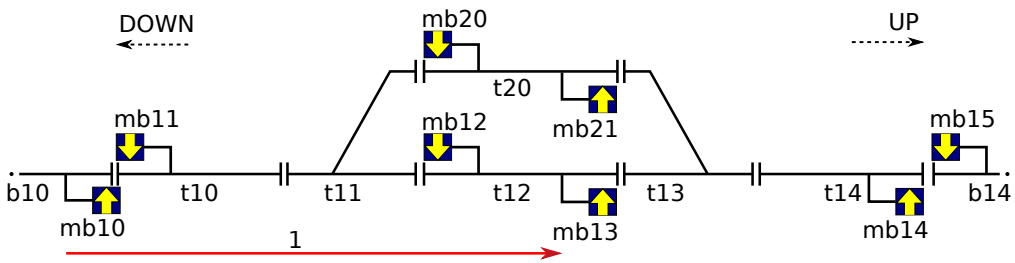
ERTMS Level 2 principles



Interlocking Tables

- An interlocking table specifying routes and conditions for *setting (reserving)* them

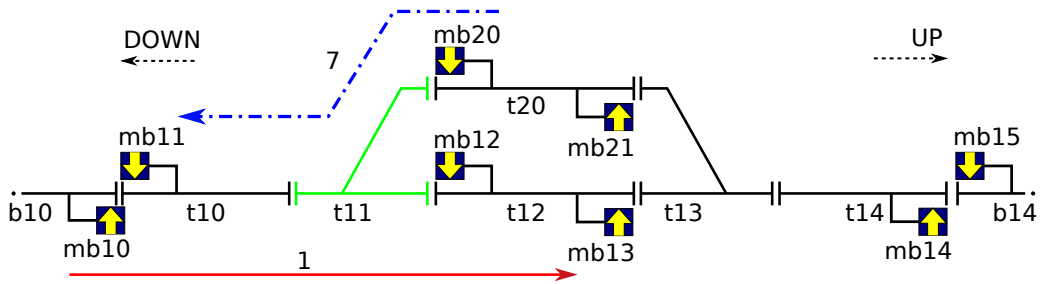
ID	src	dest	points	signals	path	conflicts
1	mb10	mb13	t11:p;t13:m	mb11;mb12;mb20	t10;t11;t12	2;3;4;5;7
2	mb10	mb21	t11:m;t13:p	mb11;mb12;mb20	t10;t11;t20	1;3;6;7;8
3	mb12	mb11	t11:p	mb10;mb20	t11;t10	1;2;5;7
4	mb13	mb14	t13:p	mb15;mb21	t13;t14	1;5;6;8
5	mb15	mb12	t11:m;t13:p	mb13;mb14;mb21	t14;t13;t12	1;3;4;6;8
6	mb15	mb20	t11:p;t13:m	mb13;mb14;mb21	t14;t13;t20	2;4;5;7;8
7	mb20	mb11	t11:m	mb10;mb12	t11;t10	1;2;3;6
8	mb21	mb14	t13:m	mb13;mb15	t13;t14	2;4;5;6



Sequential Release

- Incrementally releasing route portions that have been traversed by the associated train
  (+) concurrency level ↑ → train throughput ↑
  (−) more complex

  E.g.: `t11` can be released as soon as the train has passed it while traveling on route `1`, then `t11` can be used to set route `7`

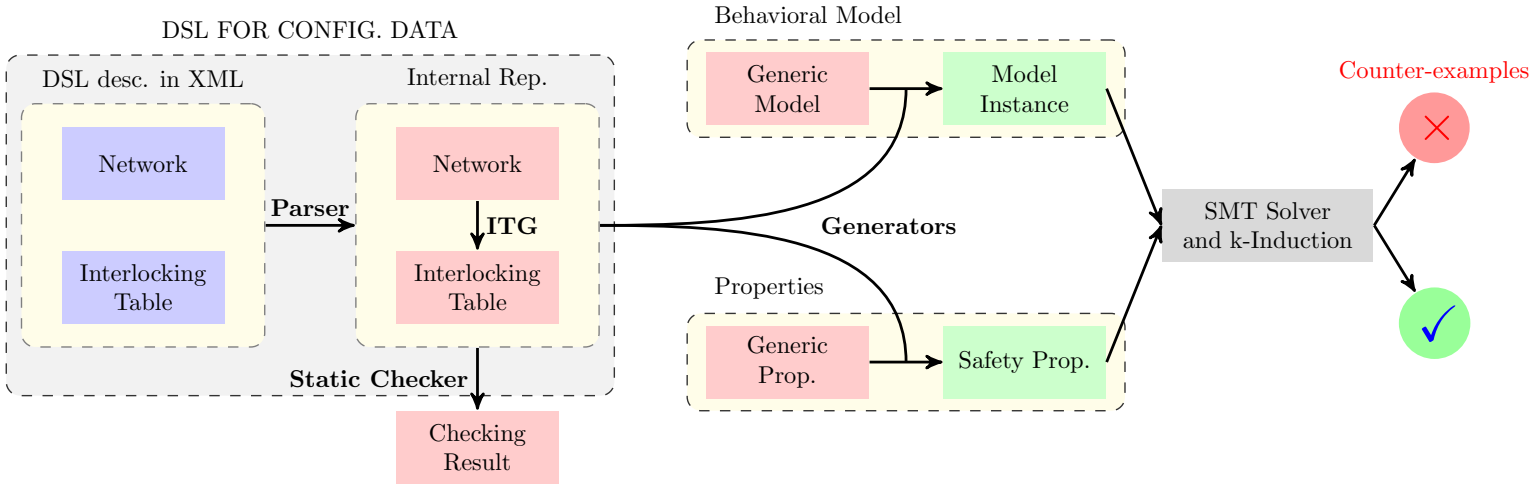


ID	src	dest	points	signals	path	conflicts
1	mb10	mb13	t11:p;t13:m	mb11;mb12;mb20	t10;t11;t12	2;3;4;5;7
7	mb20	mb11	t11:m	mb10;mb12	t11;t10	1;2;3;6

Agenda

1. Background
2. Interlocking Case Study
3. Toolchain
4. Model and Safety Properties
5. Verification Technique
6. Conclusion

Toolchain Overview



- *Reconfigurable model*:
  - Configuration data of interlocking systems (network + interlocking table)
  - Generic behavioral model and safety properties which can be instantiated with the configuration data
- *2-step verification and validation (V&V)*
  - Validate configuration data by the static checker
  - Verify safety properties for model instances: bounded model checking (BMC) + inductive reasoning
- *Types of identified errors*
  - Errors in the configuration data
  - Errors in the design of interlocking protocol
- Implemented as a tool-chain using RT-Tester toolbox and SONOLAR SMT solver[3]

[3] AG BS, University of Bremen, Germany

Generic Behavioral Model and Properties

❶ Generic behavioral model describes the behaviors of

- Interlocking controller
- Its environment: point switching, train movements

→ (instantiated with configuration data) → model instance - a Kripke structure K

$$r : Route \cdot r.MODE = FREE \wedge r.MODE' = MARKED$$

instantiated $\downarrow r = r1 \ldots r4$

$$(r1.MODE = FREE \wedge r1.MODE' = MARKED) \vee$$
$$(r2.MODE = FREE \wedge r2.MODE' = MARKED) \vee$$
$$(r3.MODE = FREE \wedge r3.MODE' = MARKED) \vee$$
$$(r4.MODE = FREE \wedge r4.MODE' = MARKED)$$

❷ Generic safety properties: no collisions, no derailments
→ (instantiated) → concrete safety properties - state invariants over variables representing vacancy status of sections

High-level Safety Properties

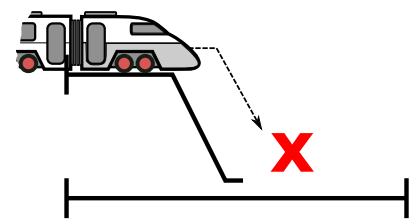
- Formulated as a state invariant ϕ over variables representing vacancy status of sections
- ϕ: *free of hazardous situations*

$$\phi \equiv \left(\bigwedge_{l:Linear} \neg Hazard_l \right) \wedge \left(\bigwedge_{p:Point} \neg Hazard_p \right) \quad (1)$$

- **Hazards**:
  (a) Head to head collision
  (b) Trains follow others collision
  (c) Derailment on points

(a) Head to head

(b) Trains follow others



(c) Derailment

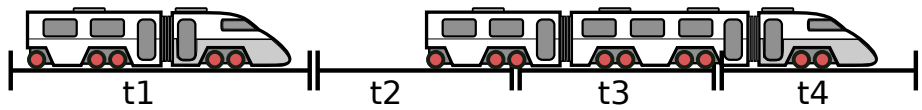
- **Proof obligation**: Prove $K \models G(\phi)$, where K is the behavioral model instance

Verification Strategy

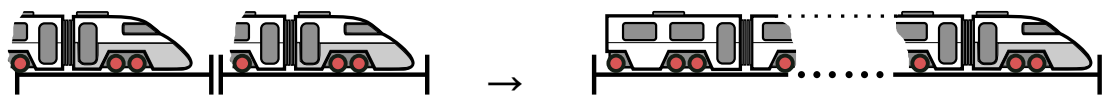
- **Strategy**: combine bounded model checking (BMC) with inductive reasoning
- To prove $K \models G(\phi)$:
  - Prove base case: ϕ holds for k consecutive states starting from the initial state
  - Prove induction step: if ϕ holds for k consecutive states starting from an *arbitrary* state, then ϕ will hold in the $(k+1)^{th}$ state.
- Base case and induction step are proved using a SMT-based model checker
- ϕ not always *inductive* → *spurious* counter-examples
  → **strengthening invariant** ψ: prove $K \models G(\phi \wedge \psi)$ instead of $K \models G(\phi)$

Strengthening Invariant Example

- *Train movement model*: distinguishes situations where the head and/or tail of the train occupy the section



→ Strengthening invariant for *train integrity*: if the *head* but not the *tail* of a train is in the *current* section, then we should find the *tail* in one of the *previous sections* (before we find another head or vacant section)



(a) × Spurious (b) ✓ Train integrity

Experimental Results

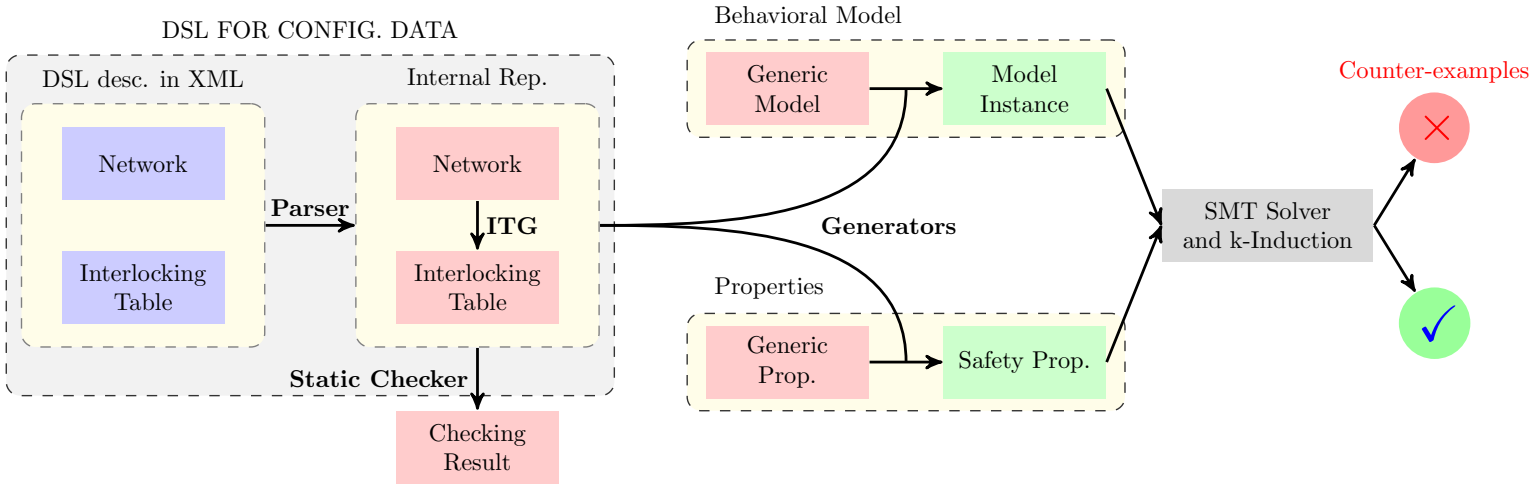
- Verified the models of interlocking systems controlling networks of realistic size, e.g. Køge st. in the *early deployment line* in the Danish Signalling Programme.
- Identified errors (if there were any) quickly in the generic behavioral model or configuration data of interlocking systems

Case	Linears	Points	Signals	Routes	Vars	Time(sec)	Memory
Toy	6	1	6	4	47	3	77 MB
Mini	6	2	8	12	66	12	139 MB
Cross	8	2	8	10	72	13	161 MB
Gt-Hd	21	5	24	33	200	154	673 MB
Køge	57	23	60	73	582	4001	4667 MB

- **How is it useful?**
  - Major number of network fractions in Denmark are smaller than Køge
  - Complex fractions, e.g. central station, can be decomposed into smaller ones, e.g. using Covering Abstraction[4]
  - Automated, gives higher level of confidence than manual verification alone

[4] Philip James et al. "Verification of Scheme Plans using CSP|| B". . In: *Software Engineering and Formal Methods*. Springer, 2014, pp. 189–204.

Conclusion



- Formal model of the forthcoming Danish interlocking systems
  - ETCS Level 2 compatible: *virtual signal* concept → handle assignment of movement authorities in the similar way as physical signals are used
  - Accommodate sequential release → more complex model
- *Pushed the applicability bounds* of FM in verifying interlocking systems further by
  - Encodings of state space, transition relation, and safety properties → can be efficiently evaluated by SMT solvers
  - Verification technique of combining BMC and inductive reasoning
- Implemented the toolchain and successfully verified the configuration data of interlocking systems controlling networks of realistic size.

Thank you for your attention!