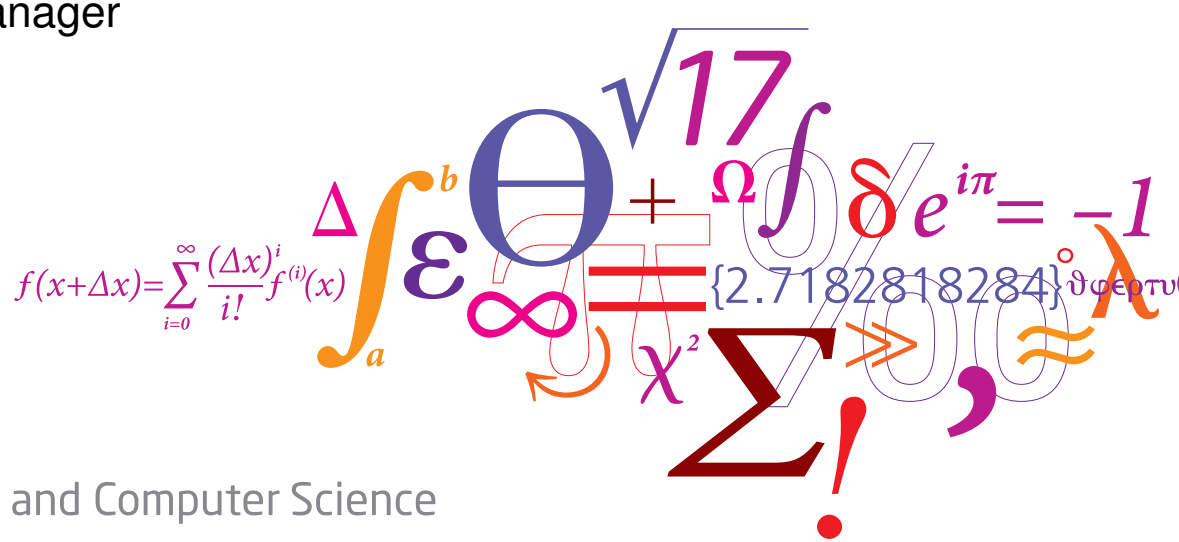
Compositional verification of railway interlockings: comparison of two methods

Alessandro Fantechi[1], Gloria Gori[1], Anne E. Haxthausen[2], and Christophe Limbrée[3]

[1] DINFO, University of Florence

[2] DTU Compute, Technical University of Denmark

[3] Belgian Railway Infrastructure Manager



UNIVERSITÀ DEGLI STUDI FIRENZE
DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DTU Compute
Department of Applied Mathematics and Computer Science

Outline

1. Introduction

2. Two compositional verification methods

3. Comparison of the two methods

4. Conclusion

The challenge of formal verification of interlocking systems

- Model checking has raised the interest of railway signaling industries, being the most lightweight from the process point of view, and being rather promising in terms of efficiency.
- Interlocking systems called for a direct application of model checking, since required safety properties *(no_collision, no_derailment,...)* are ”easily” expressed in temporal logic.
    - **No collisions:** *Two trains must never occupy the same track section at the same time.*
    - **No derailments:** *A point must not be switched, while being occupied by a train.*
- However, due to the high number of boolean variables involved, automatic verification of sufficiently large stations typically incurs in combinatorial state space explosion problem.
- SAT/SMT-based verification is currently the most promising option and is used in industrial solutions.
- Nevertheless, verification of large interlocking systems is still a challenge and decomposition can help to face it.

Different methods, common aim

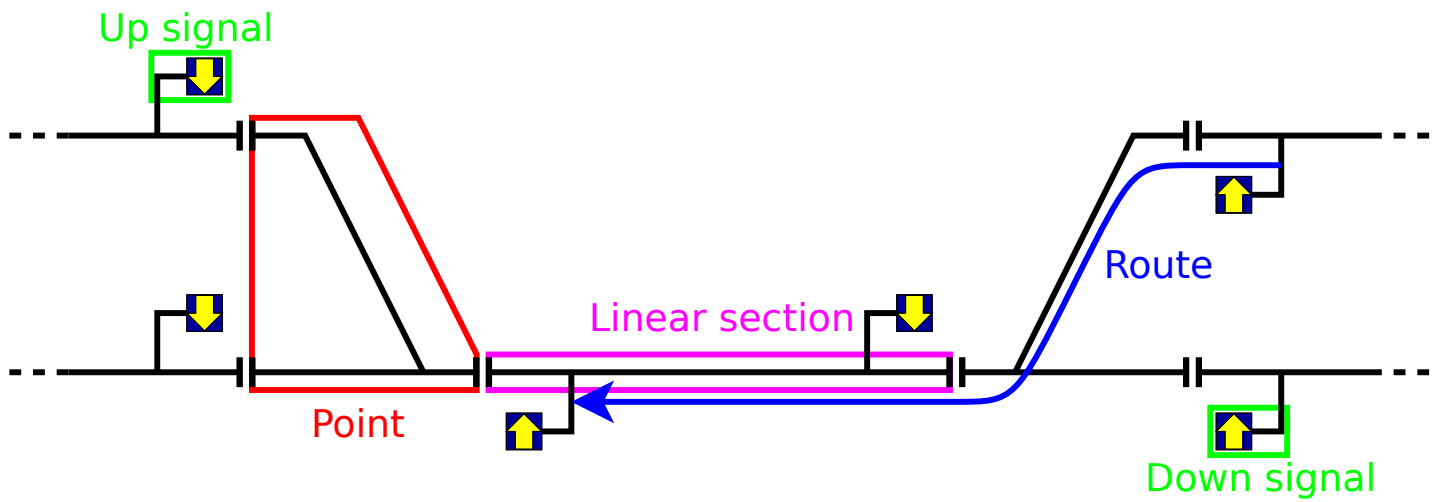
- Automatic verification is possible with smaller networks $\rightarrow$ Decompose a network can help.
- We compare two decomposition methods which differ w.r.t.:
  - cut methodology;
  - verification strategy and tools.

Objective

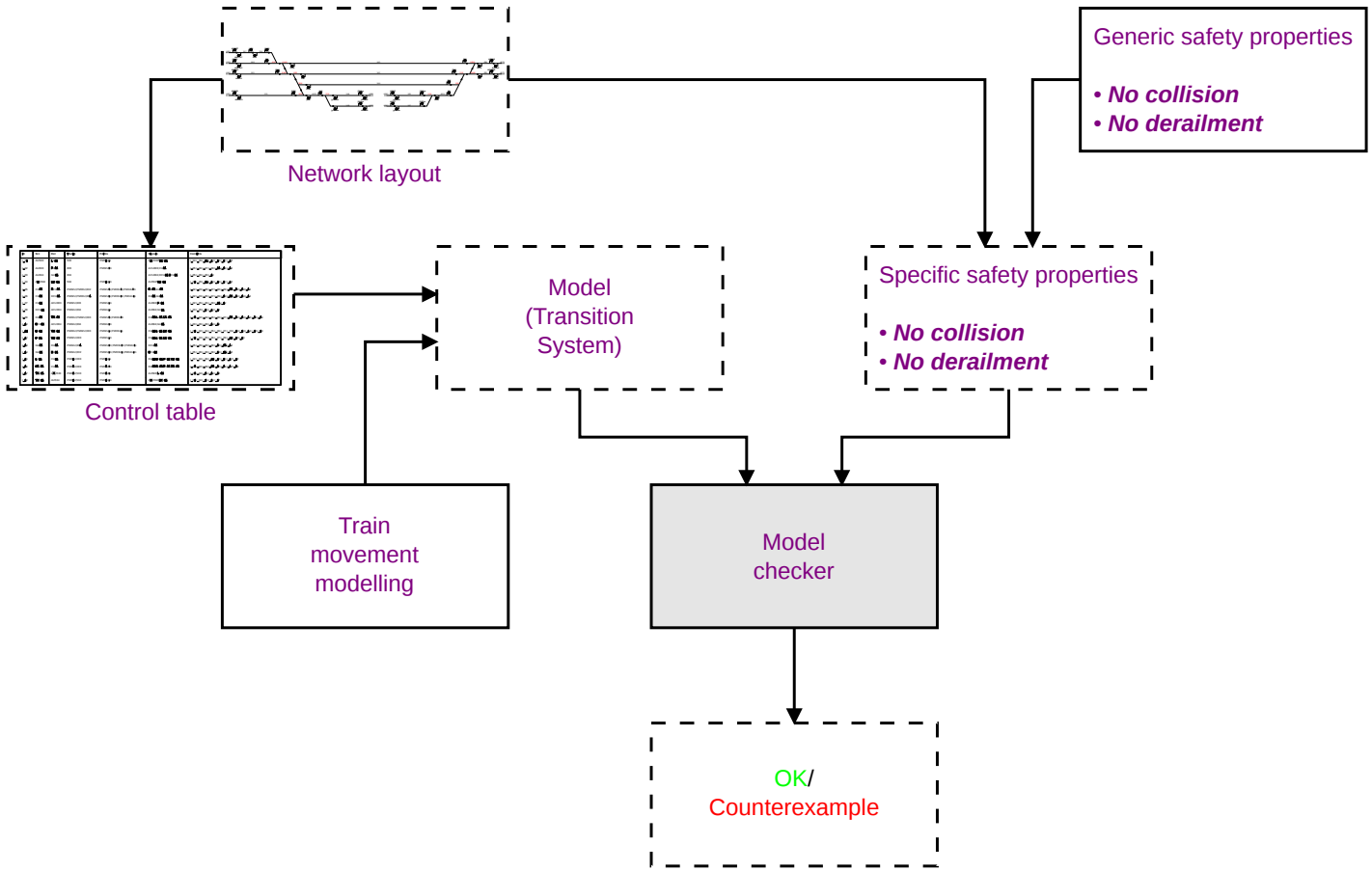
Highlight differences, similarities, pros and cons depending on different factors that may impact.

Interlocking Systems

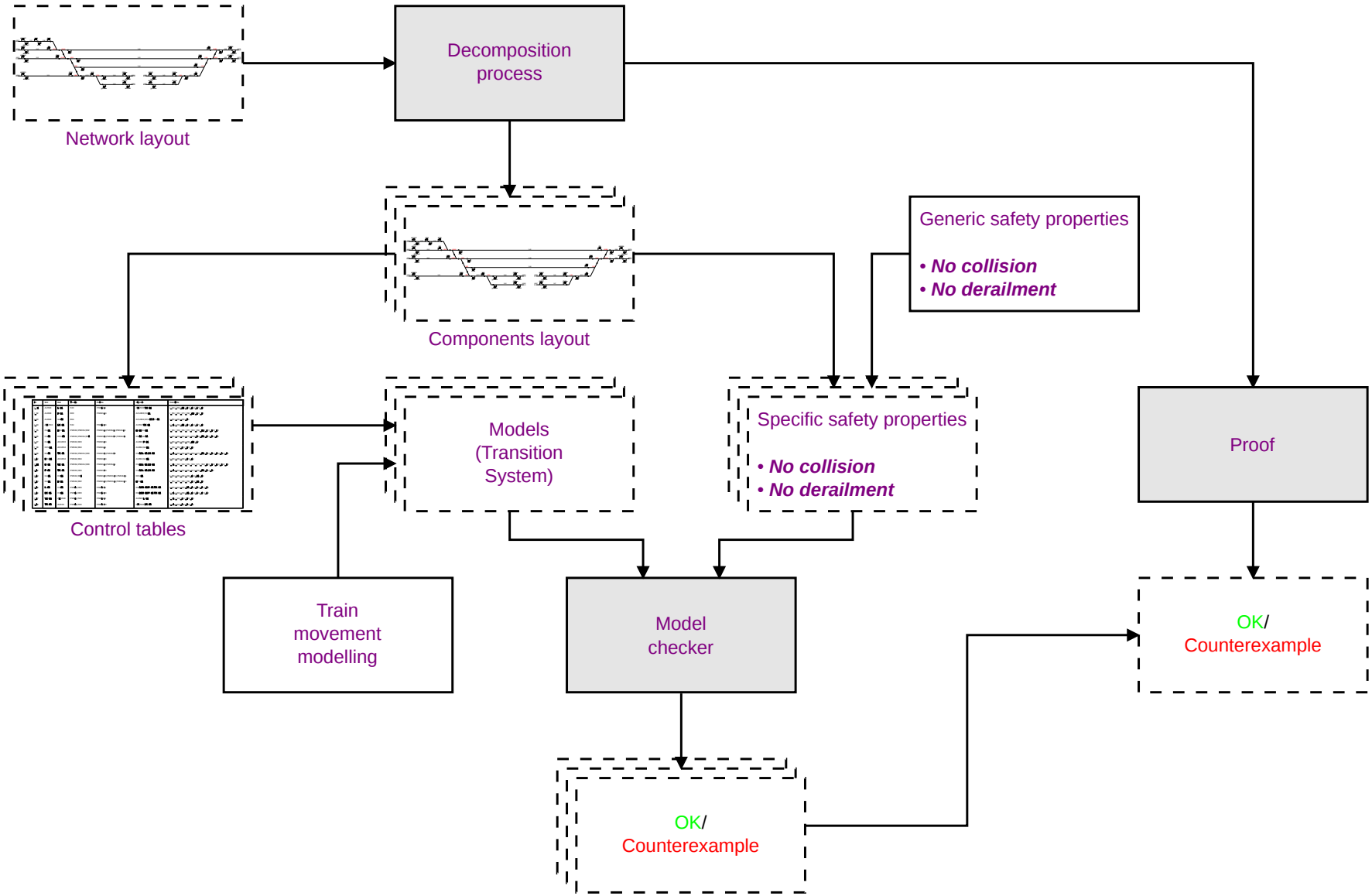
- Receives *route requests* from the traffic control center.
- Sets (that is, reserves for exclusive use by a train) a requested route, if no conflicting route is already set.
- While setting a route it orders the points to be locked in correct positions for the route.
- Once a route is set, it should be “signalled” to the train by setting a signal to “PROCEED”!
    - In ERTMS/ETCS level 2, signals are *virtual*, and are replaced by static *markerboards* and radio communication.
- Once a train enters the route, it sets the (virtual) signal to CLOSED.
- It releases the route for further use by other trains, when the train has finished using it.



Monolithic Verification Process

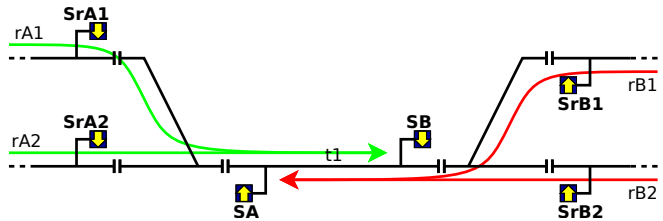


Compositional Verification Process



RobustRailS Specification = Network + Route Table

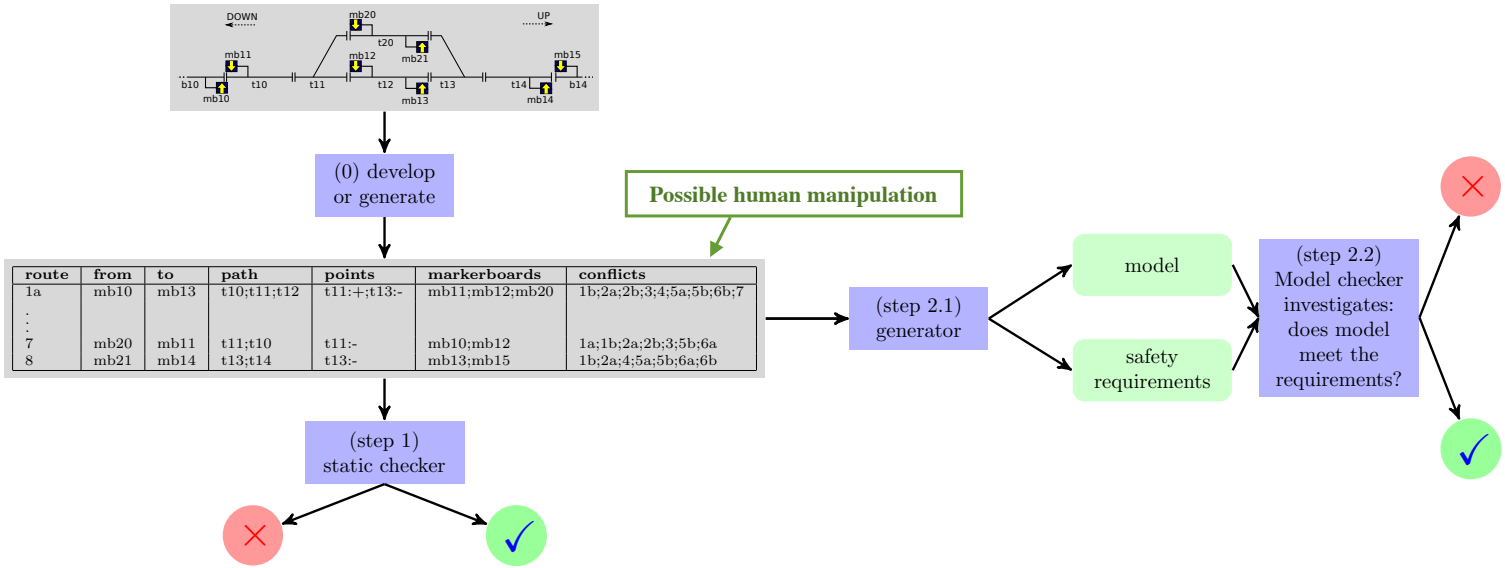
Network



Route table

id	src	dst	path	points	signals	conflicts
rA1	SrA1	SB	t1	pm1:m	SrA1;SrA2;SB	rB1;rB2;
..	...	...	...	...	...	...
rB1	SrAB1	SA	t1	pm2:m	SrB1;SrAB2;SA	r1;rB2;

RobustRailS Verification Method & Tools[1]



A two step verification:

1. The *static checking step* is used to find errors in the control table.
2. The *model checking step* is used to find errors in the control algorithms of the instantiated system model.

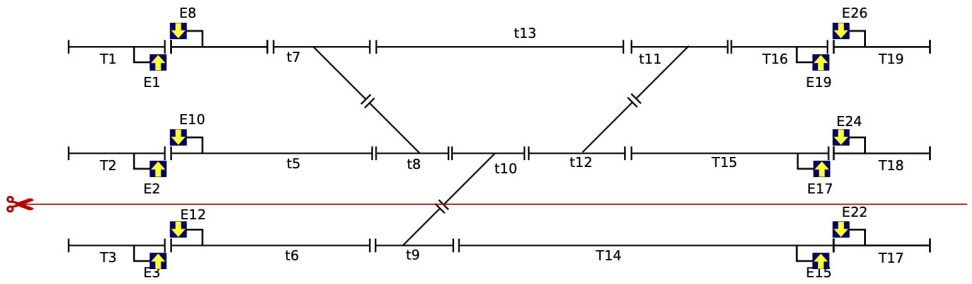
[1] The method and tools were developed by Anne Haxthausen, Jan Peleska and Linh H. Vu in collaboration with the Danish railways in the RobustRailS project, 2012-2017, `http://www.robustrails.man.dtu.dk/`

Louvain Verification Method

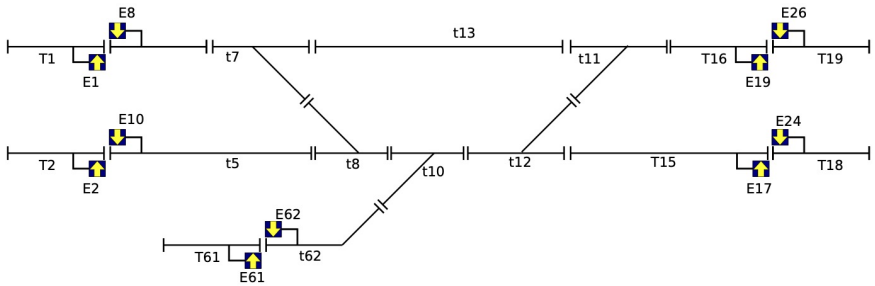
The Louvain verification method exploits a set of tools to automatically verify safety properties on a railway interlocking system model generated from the application data.

1. Generate a model of the RIS based on its application data.
2. Generate a model of the train and the safety properties applicable to a specific network layout.
3. Combine the models of the interlocking with two instances of the train in a SMV model and verify the properties with nuXmv.

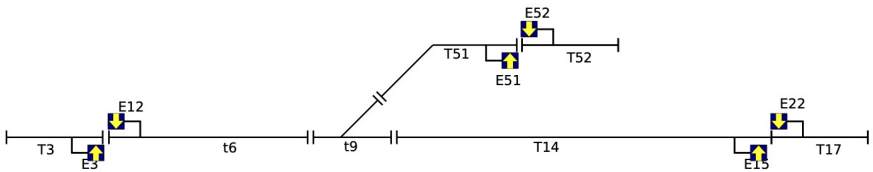
RobustRailS: compositional method



The total network and the cut.



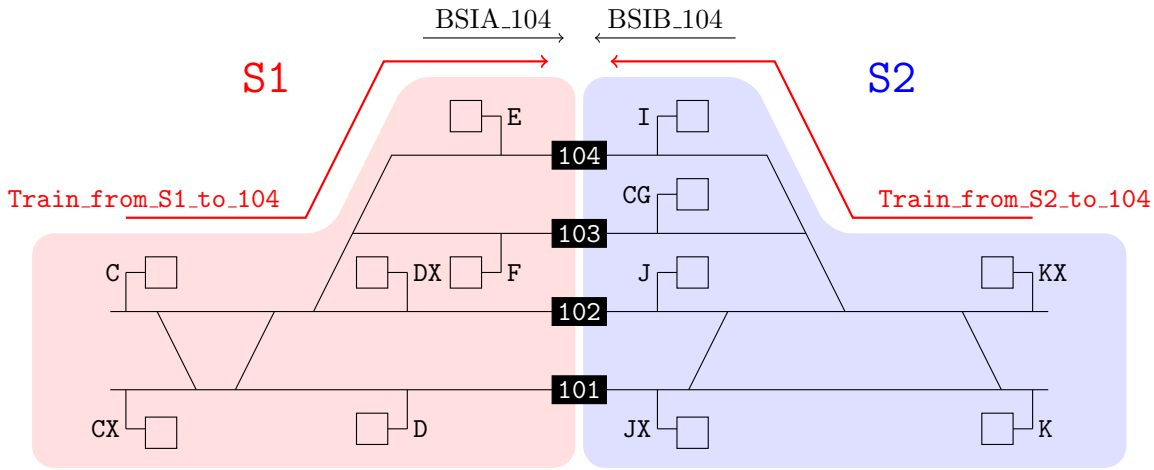
The *high* sub-network.



The *low* sub-network.

1. Cut the network into n subnetworks applying allowed network cuts.
2. For each subnetwork use the RobustRailS tools verification steps described before.

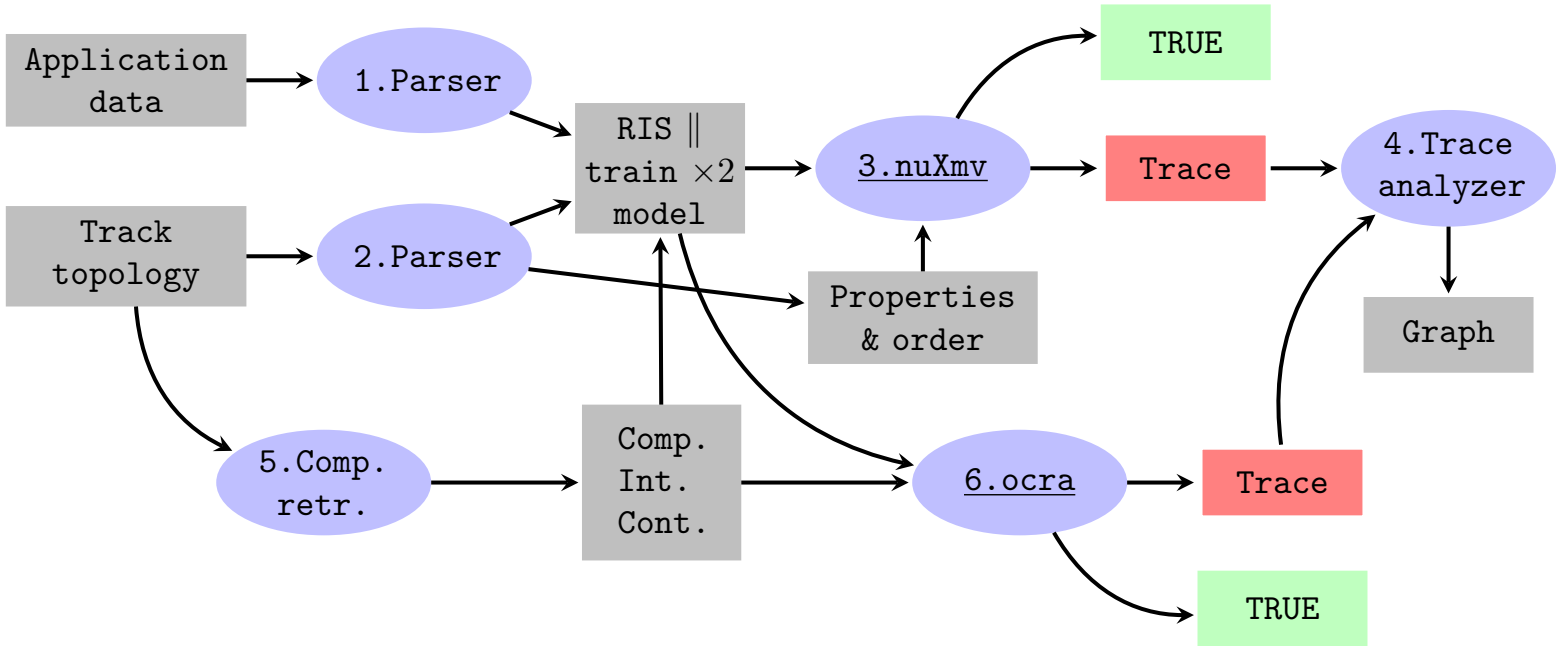
Louvain: compositional method



Application data

```
*Q_R(C_104)
  if R_C_104 a , R_C_104 xs
    P_02BC cfr, P_01AC cfn , P_03C cfn
    ...
  then R_C_104 s
    P_02BC cr , P1_02BC cr , P2_02BC cr
    U_C_14C l, U_14C_15C l, U_15C_EC  l
    if U_BSIB(104) f then U_BSIA(104) l
```

Louvain Verification Method & Tools[2]

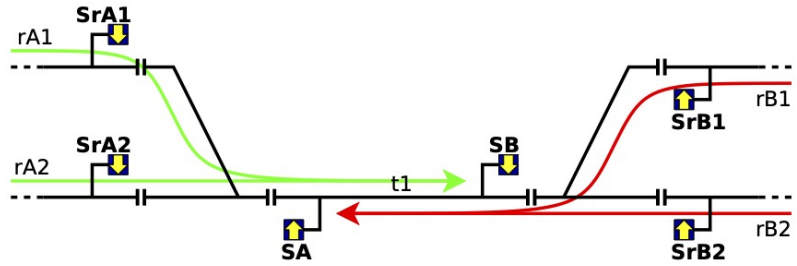


A two steps verification:

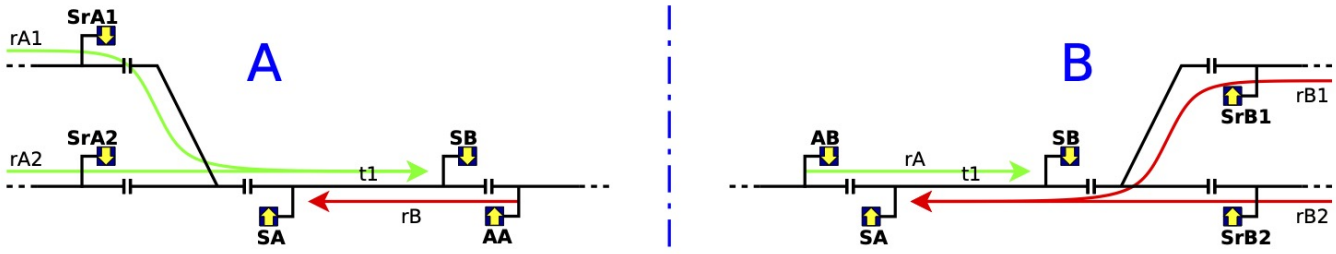
1. *OCRA* is used to find errors in the interfaces between different networks.
2. *nuXmv* is used to find errors in the implementations (models) of the RIS.

[2] The method and tools were developed by Christophe Limbrée in the context of his PhD Thesis.

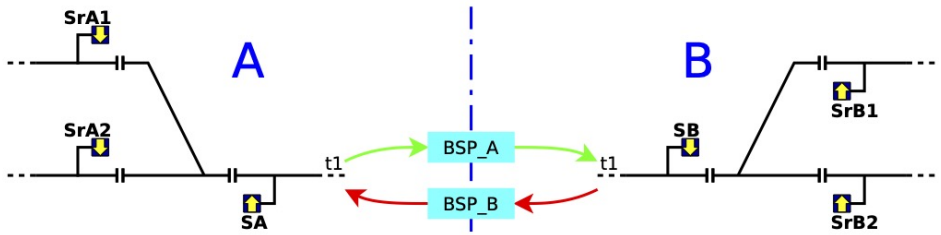
RobustRails vs. Louvain decomposition



(a) Example station layout.



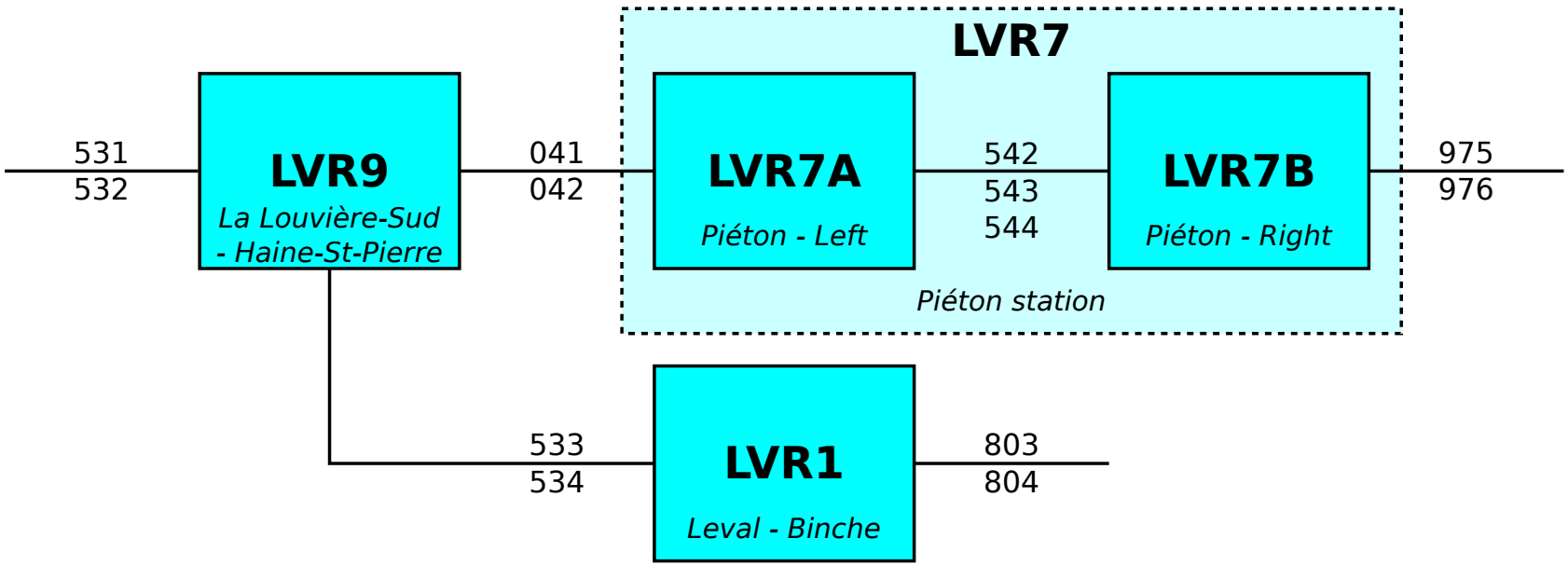
(b) RobustRailS cut implementation.



(c) Louvain cut implementation.

Case study

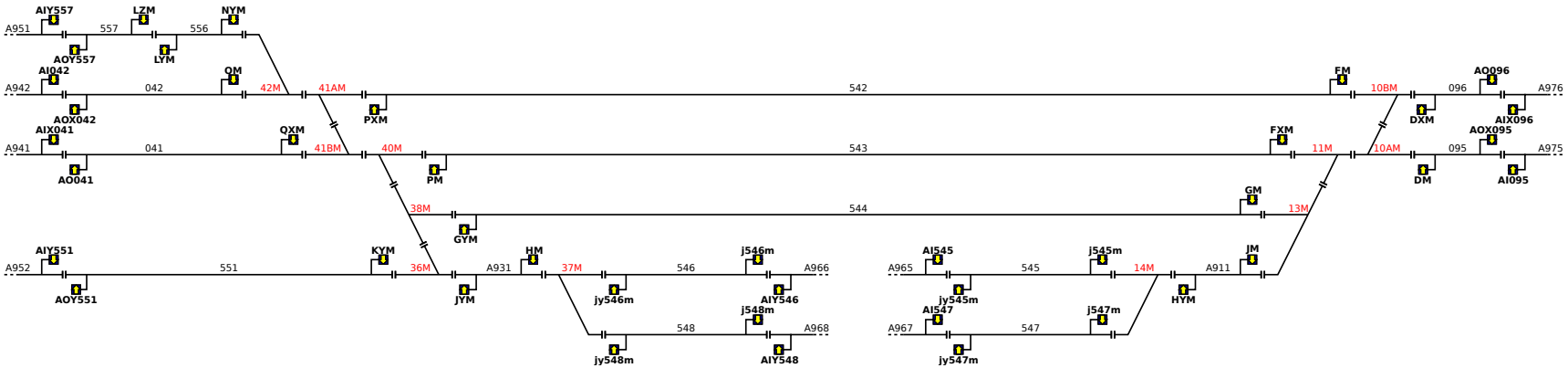
Network components topology



Components Characteristics

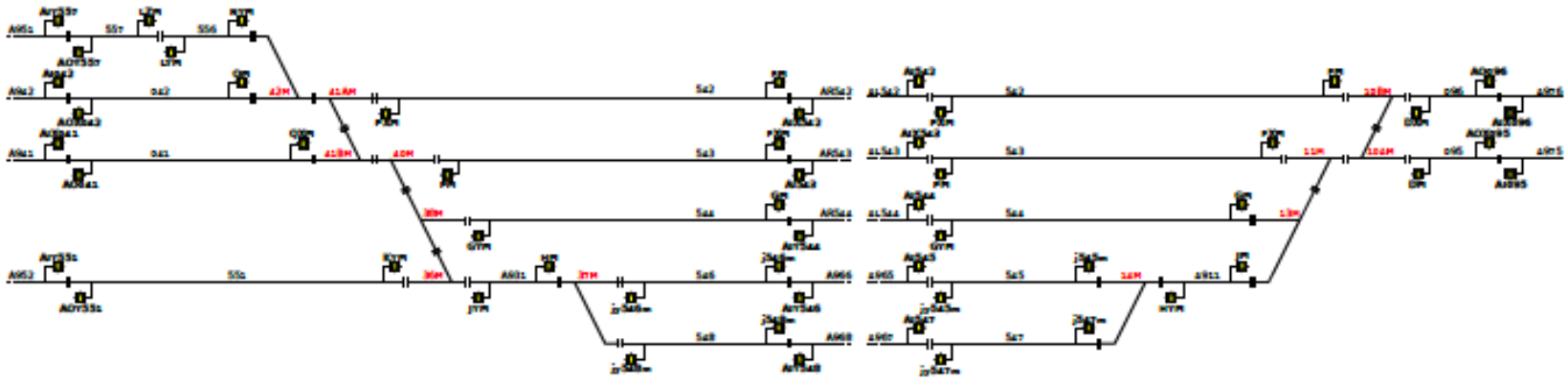
id	routes	points	signals
LVR7	48	13	19
LVR7A	30	8	11
LVR7B	18	5	8
LVR1	14	4	10
LVR9	12	4	6

LVR7 (Piéton)



The topology of the station makes the cut possible and the number of routes (48) makes the advantages of the compositional method more evident.

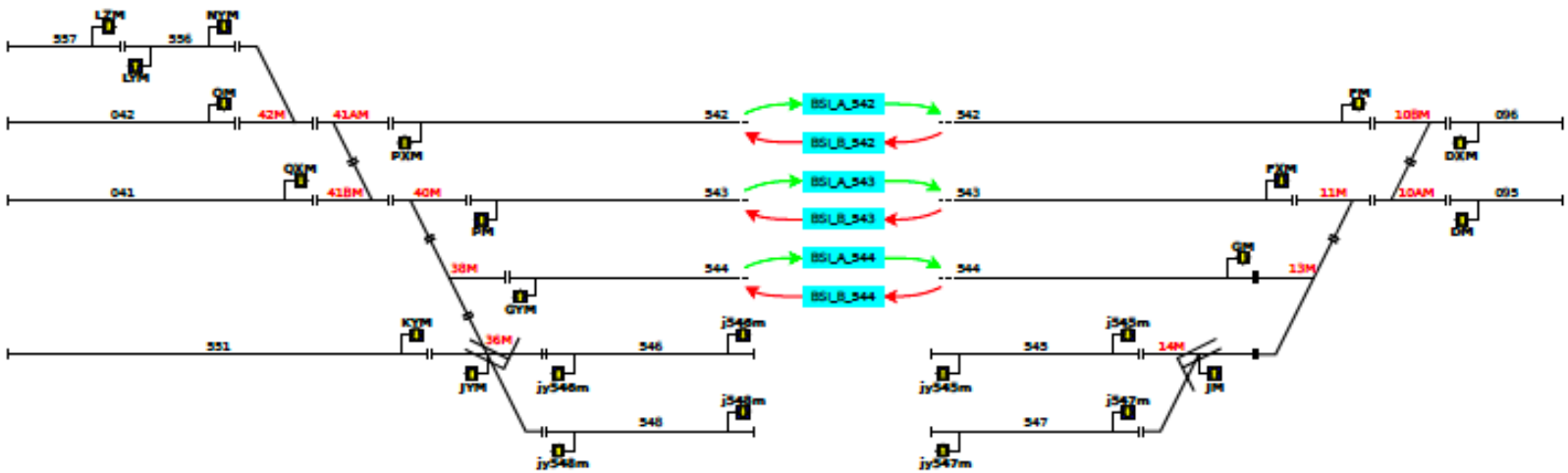
LVR7 (Piéton): RobustRailS decomposition



(a) Piéton left component.

(b) Piéton right component.

LVR7 (Piéton): Louvain decomposition



Experiments results

Time

id	routes	RobustRailS (s)	Louvain (s)
LVR7	48	2387	Not feasible
LVR7A	30	670	17670
LVR7B	18	108	5540
LVR1	18	38	532
LVR9	18	33	4436

Memory

id	routes	RobustRailS (MB)	Louvain (MB)
LVR7	48	5467	Not feasible
LVR7A	30	2083	152
LVR7B	18	846	125
LVR1	18	413	48
LVR9	18	415	81

Lessons learned

- The actual verification performance depends on many factors that differ in the two methods.
- The results on the verification of Piéton (LVR7) clearly show the advantages of the compositional verification process in both methods.
- The experiments on LVR9 show that for components with low internal complexity, the number of interfaces to be verified has a higher impact on the verification time and memory for Louvain method.
- RobustRailS provides a tool performing automatic decomposition from a cut specification. The Louvain method takes advantage of existing sub-networks definition in Belgian signaling principles and provides a fully automatic decomposition.
- Both methods provide the possibility of running verification of the sub-networks in parallel on multi-core machines.
- Models generated automatically for verification tools.
- Amount of generated proof obligations to prove the safety of the decomposed networks:
    - For the RobustRailS compositional method the soundness of the component verification has been proved a priori (once-and-for-all).
    - The Louvain method needs additional verification obligations: the verification of the contracts related to its interfaces.

Future Work

- For the Louvain approach:
  - Apply decomposition by functions rather than based on the topology.
  - Integrate the flank protection properties between sub-networks.
- For the RobustRailS approach:
  - Extend the cut tool to automatically find optimal places to cut.
- Study the applicability of the decomposition approaches to different verification frameworks and tools including the hybrid approach.
- Further comparison between the two approaches extending the reasoning to the other four kind of interfaces in Belgian signaling principles.

Questions?