Safety Analysis of a CBTC System: A Rigorous Approach with Event-B

Mathieu Comptier, David Déharbe, Julien Molinero Perez, Louis Mussat, Thibaut Pierre, and Denis Sabatier
RSSRail – International Conference on Reliability, Safety and Security of Railway Systems: Modelling, Analysis, Verification and Certification
November 15th 2017
Pistoia, Italy

Introduction

- Feedback on the safety analysis conducted on the CBTC Octys, a RATP product.

- Characterized by a rigorous approach supported by lightweight use of formal methods.

- Output: safety analysis spanning a set of related documents that provide:
  - a logical argument establishing the safety properties guaranteed by Octys
  - a set of sufficient requirements (hypotheses)
  - hypotheses can be used as
    - rules for data validation on lines equipped with Octys
    - proof goals for sub-system designs
  - the argument builds a theory useful to guide future developments of Octys

Overview

- CBTC
  - Generalities
  - Octys
- Mathematically grounded safety analysis
- Elements of methodology
- Grounded safety analysis in action
- Example: Track circuits backup
- Lessons learnt and prospective

Communication-Based Train Control

- Distributed system
  - e.g. carborne controller, zone controller
  - distribution of responsibilities varies according to CBTC

- Increase throughput
  - reduce headway
  - substitute failing interlocking devices

- Improve safety
  - continuous spacing control
  - passenger transfers

- Reduce wayside signaling costs

Octys: Industrial Context

≡ Octys = Open Control of Trains, Interchangeable and Integrated System

≡ RATP modernization program for 13 metro lines with drivers.

- brownfield deployment

≡ Specific challenges:

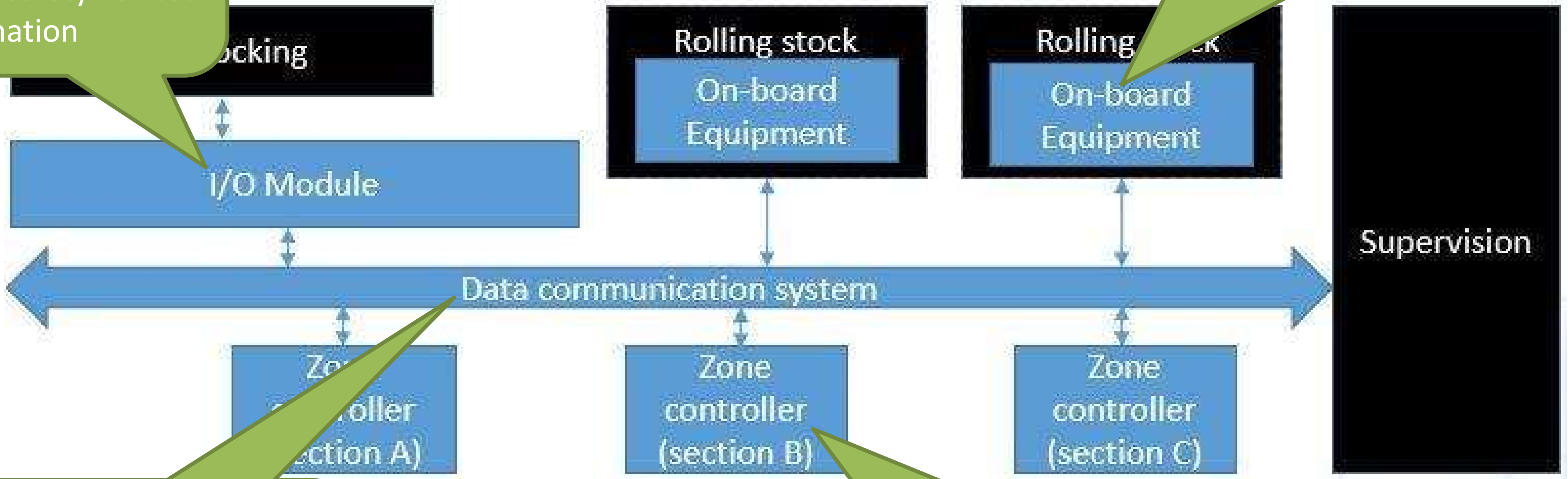
- no disruption of service
- multi-sourcing and interoperability

≡ Already deployed on several RATP lines

- Multi-line and multi-vendor: suitable for a formal safety proof

- Interface zone controller with interlocking
- Treats safety-related information

- Estimate and send position (cartography, beacons, odometry)
- Calculate and control max speed from MAL



- Distributed among equipments
- Protocol comprises clock synchronization

- Track trains on the line (positions, interlocking sensors)
- Calculate and send MAL to equiped trains

Methodology - Organization

- System operator – RATP:
  - 2-3 staff
  - expertise in formal methods and railway systems
- Safety analysis team – ClearSy:
  - 3-5 engineers
  - expertise in formal methods
  - different backgrounds (including railway systems)
- Solution expert – Siemens:
  - 1 staff with deep technical knowledge of Octys

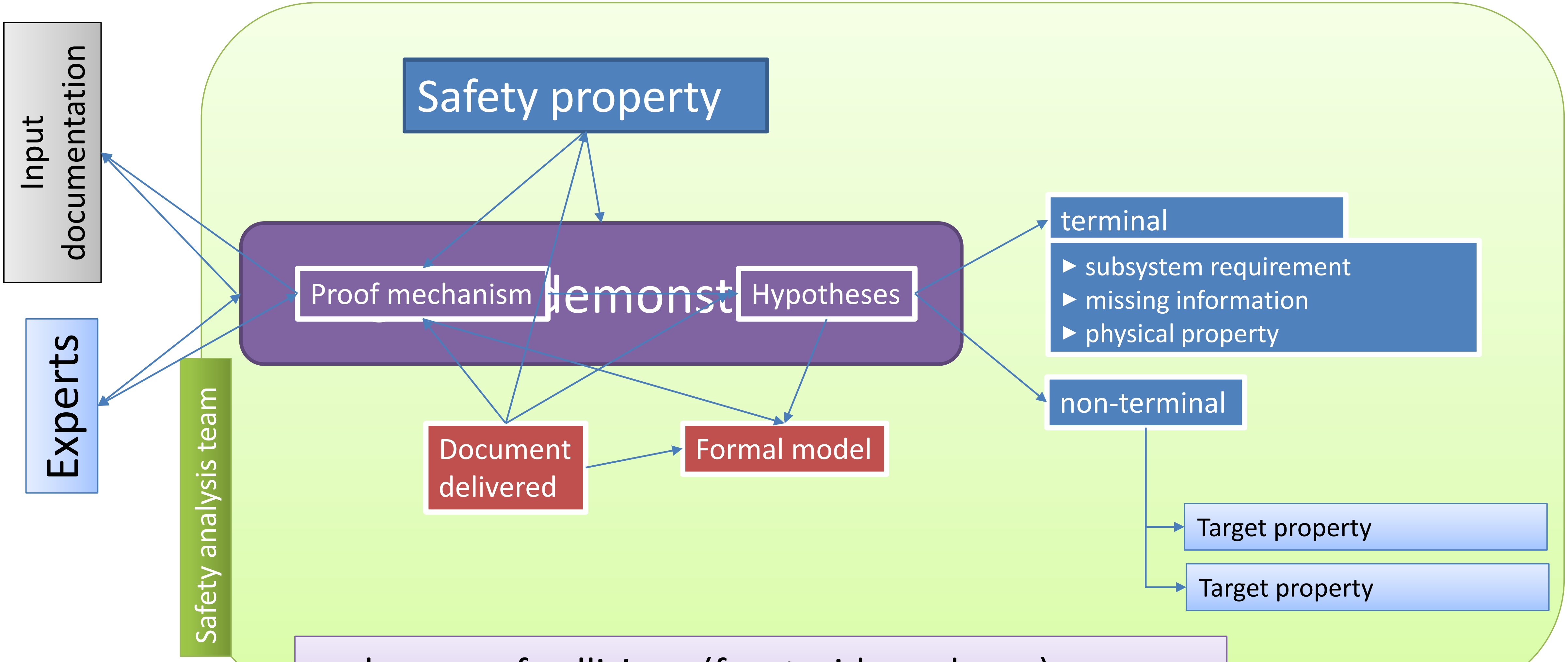
Methodology - Input

- System technical specification
- System functional specification
- System base data specification
- System parameter description
- Spec external interface signalling
- Interface signalling design rationale
- ZC functional specification
- TC functional specification
- IO module functional specification
- Inter equipment protocol description
- IO module/ZC interface
- ZC/TC interface

Methodology: communication

- Monthly meetings, ad hoc communications by email and phone
- Discussion of specific Octys functionalities – presentation of analysis of properties
  - Discussion based on scenarios
    - Clarify understanding of functionalities
    - Focused technical questions
  - Presentation of safety analysis
    - Validation of hypotheses
    - Description and validation of proof mechanism
- Before meetings with partners, the safety analysis team makes internal presentations to consolidate the arguments.

Approach



- absence of collisions (front, side and rear)
- absence of derailments over uncontrolled switches

Formal verification of proof mechanisms

- Each proof mechanism is validated with *tools*: Event-B and Atelier B.
- Event-B provides a computerized mathematical language to formalize systems.
- Event-B requires that the consistency and well-definedness of the formalization is proved using *pure logic and mathematical reasoning*.
  - Event-B has no built-in knowledge of railway systems or CBTCs
  - Each hypothesis needs to be thoroughly formalized with mathematical objects.
  - No reasoning shortcut (or error!) is possible.
- Atelier B is a toolset to write Event-B models and to perform all such proofs.
- *Beware*: Event-B is *not* a mean to *find* proof mechanisms.
  - But we find it useful for their consolidation and *necessary* for their verification.

Practical Application of Approach to Octys

- Octys is a CBTC for deployment on lines already equipped with an interlocking system.

- Neither the CBTC nor interlocking guarantee safety by itself.
  - Octys has not been designed to prevent front and side collisions, derailment on unlocked switches.
  - interlocking does not prevent rear collisions between equipped trains

- The safety analysis must take into account the interplay between the CBTC and interlocking.

Top-level argument (in a nutshell)

≡ identify a protection zone for each kind of train

$pz : TRAIN \rightarrow PZ$

≡ such that each train stays in its protection zone by its braking forces:

$\forall tr \cdot tr \in TRAIN \Rightarrow tr \subseteq pz(tr)$

- define protection zone based on terrain objects
- model any possible terrain objects change as an evolution of protection zones

≡ no geometrical intersection between protection zones:

$\forall tr_a, tr_b \cdot tr_a \in TRAIN \wedge tr_b \in TRAIN \wedge tr_a \neq tr_b \Rightarrow pz(tr_a) \cap pz(tr_b) = \emptyset$.

≡ anti-collision property:

$\forall tr_1, tr_2 \cdot tr_1 \in TRAIN \wedge tr_2 \in TRAIN \wedge tr_1 \neq tr_2 \Rightarrow tr_1 \cap tr_2 = \emptyset$.

≡ in a protection zone, switches are locked:

► similar reasoning applies

Top-level argument (in a nutshell)

- to model evolutions on concrete objects corresponding to protection zone, we formalize evolutions on protection zones.

- example: if the protection zone of train tr_1 is extended by a track portion new:

  $pz(tr_1) \leftarrow pz(tr_1) \cup new$
- then we have to argue that all the properties stated before still hold.

- For example

  $\forall tr_a, tr_b \cdot tr_a \in TRAIN \land tr_b \in TRAIN \land tr_a \neq tr_b \Rightarrow pz(tr_a) \cap pz(tr_b) = \emptyset.$
- We have to prove :

  $$\forall tr \cdot tr \in TRAIN \land tr \neq tr_1 \Rightarrow (pz(tr_1) \cup new) \cap pz(tr) = \emptyset.$$

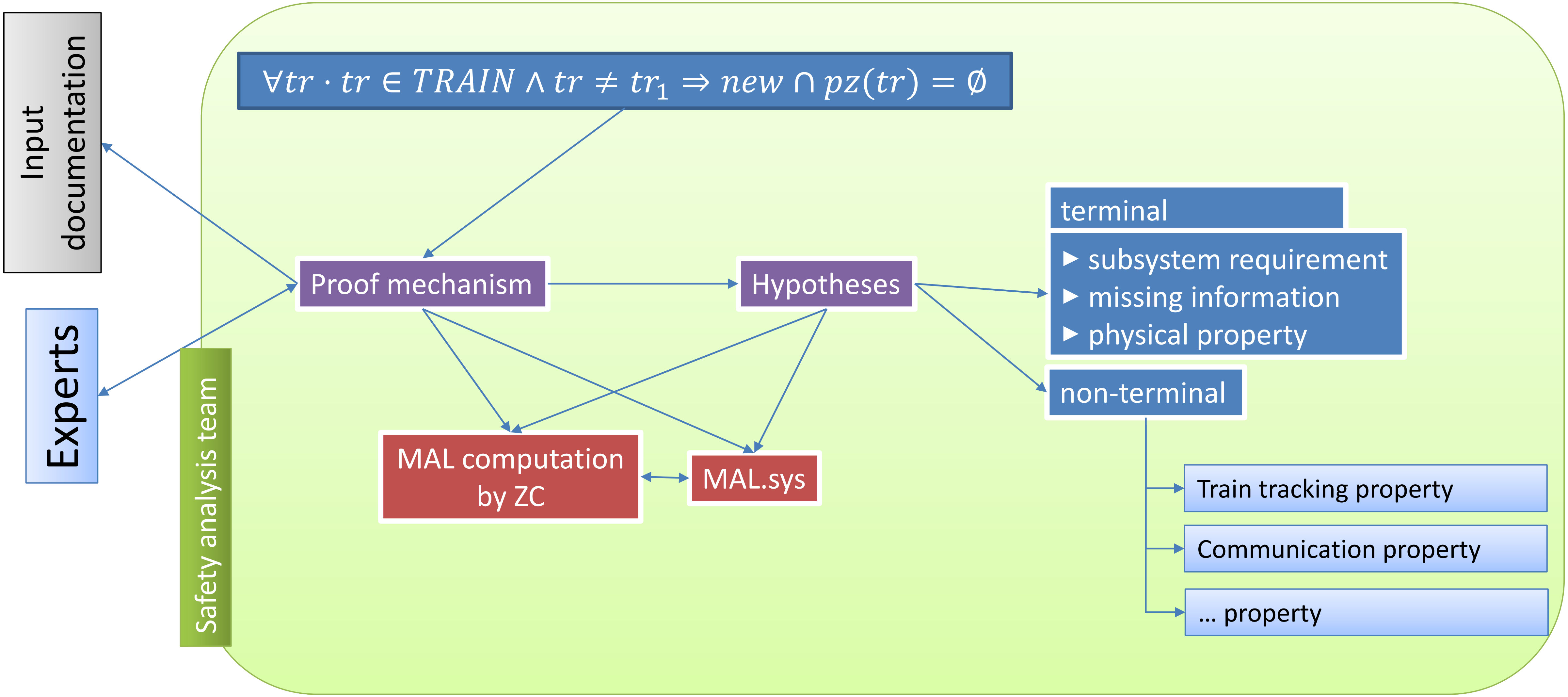
  Since

  $$\forall tr \cdot tr \in TRAIN \land tr \neq tr_1 \Rightarrow pz(tr) \cap pz(tr_1) = \emptyset,$$

  it is sufficient to prove :

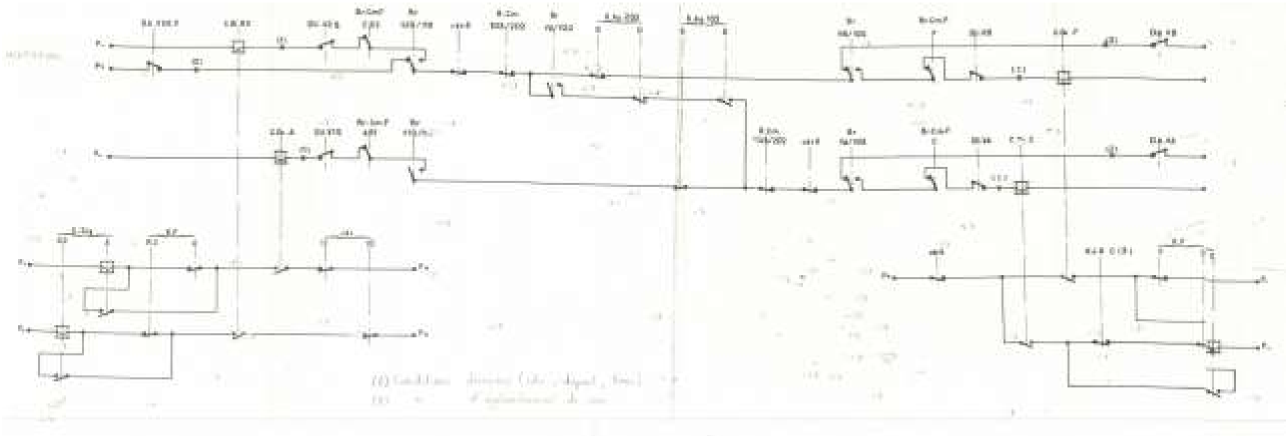
  $$\forall tr \cdot tr \in TRAIN \land tr \neq tr_1 \Rightarrow new \cap pz(tr) = \emptyset.$$

Decomposition in action



Practical Application of Approach to Octys

- Initial plan: analyze all possible evolutions of protection zones regarding interlocking and CBTC functions
  - issue: access only to documentation of CBTC
  - sole hypothesis on interlocking: safe before deployment of CBTC
- Experience shows this approach requires exposing more details on interlocking, which we decided to avoid.
  - cost: interlocking circuits are complex devices



  - usability: we cannot forecast the interlocking circuits of lines where Octys will be deployed in the future

Practical Application of Approach to Octys

- Final argument: based on protection zones, as usual.
- Protection zones are given a precise semantics in terms of concrete objects (signals, train envelopes and rollback, etc.)
- All possible evolutions of these concrete objects are modelled as evolutions of protection zones
  - identify all properties on equipped trains that need proof
  - identify hypotheses on protection zones from interlocking
- Interlocking-related hypotheses on protection zones have to be verified by interlocking experts based on
  - properties of the CBTC we provide them
  - properties of interlocking these experts identify and verify

Example: track circuit backup

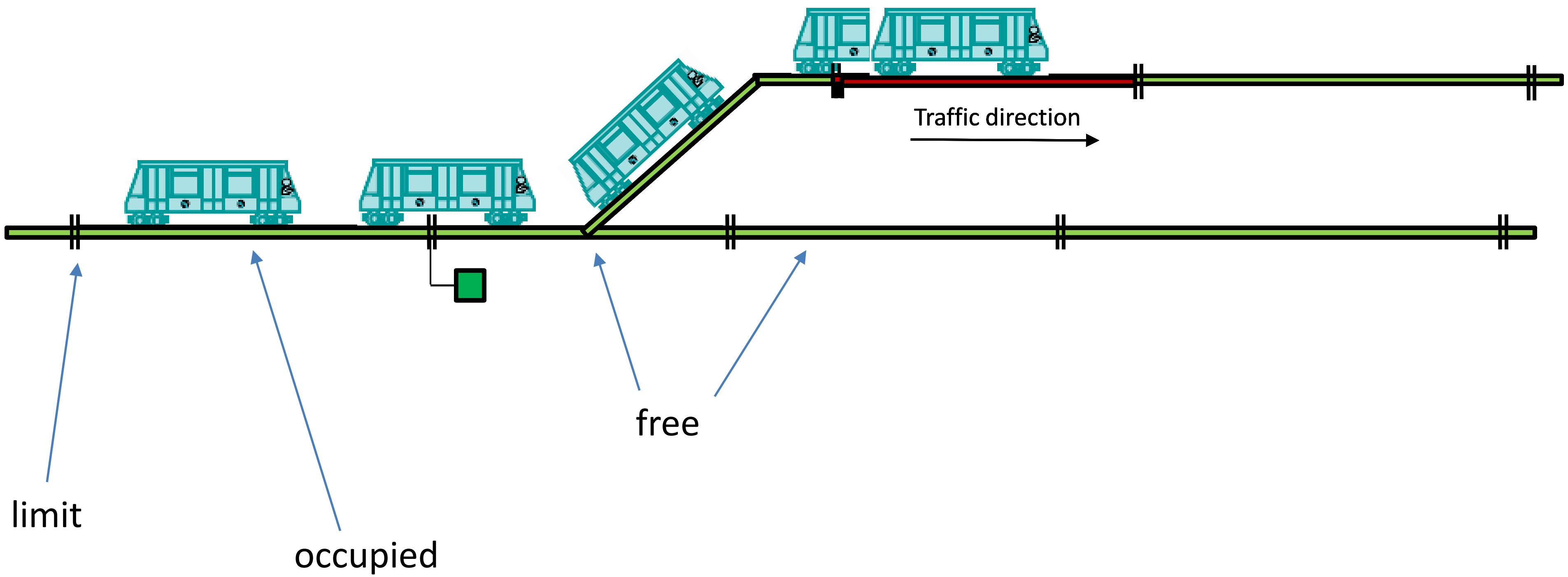
« *We delegate to interlocking experts the verification of interlocking hypotheses, providing them properties of the CBTC*. »

- Octys includes functionalities altering interlocking inputs, e.g. *track circuit backup*.
- Track circuits are train detection devices.
- Top-level argument:
  - safety properties are guaranteed by the existence of protection zones.
  - requirement for interlocking: existence of such protection zones against
    - front and side collisions and
    - derailment on unlocked switches
    - rear collisions for non-equiped trains
- Informated guess: interlocking protection zones involve track circuits.

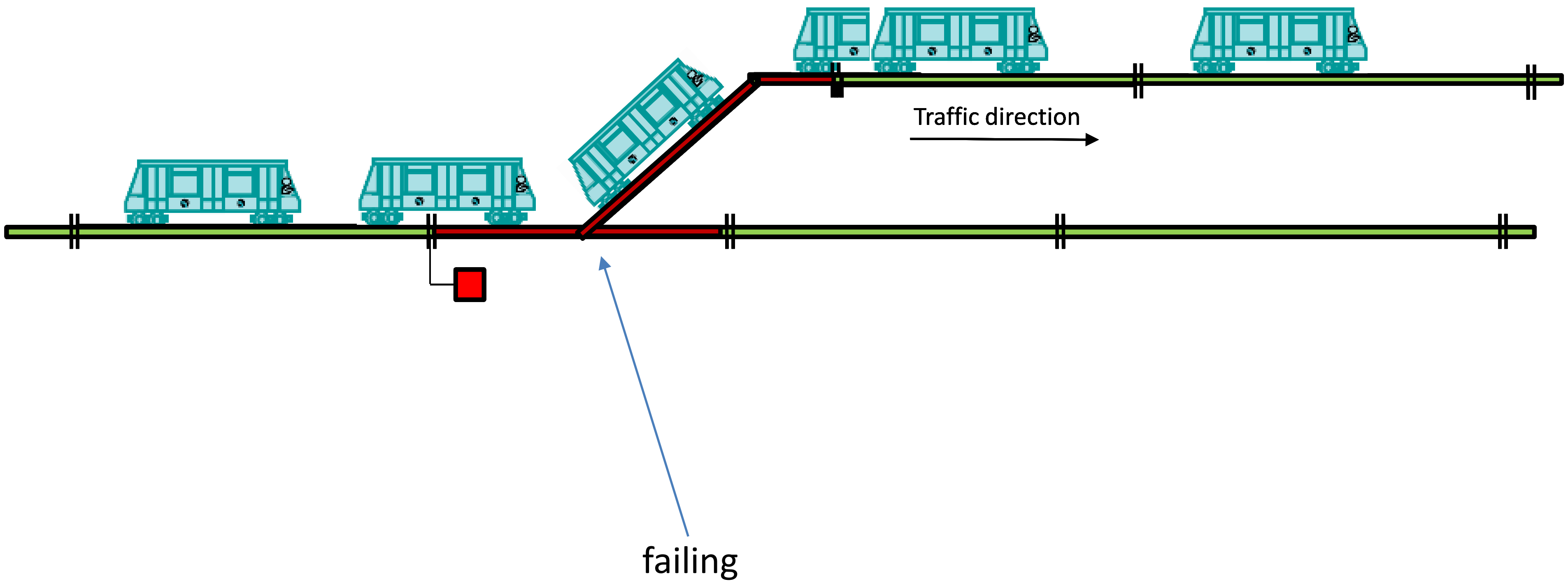
Example: track circuit backup

- It is our duty to inform interlocking experts that track circuit behavior is changed.
- Interlocking cannot rely on the classical physical properties of track circuits to guarantee its protection zones.
- First contribution: identify and prove properties on track circuits that we estimate sufficient for interlocking experts to guarantee the protection zones.
  - in collaboration with an interlocking expert
- Second contribution: find the argument establishing these properties hold.
  - formalize it and prove its correctness.

Scenario without track circuit backup



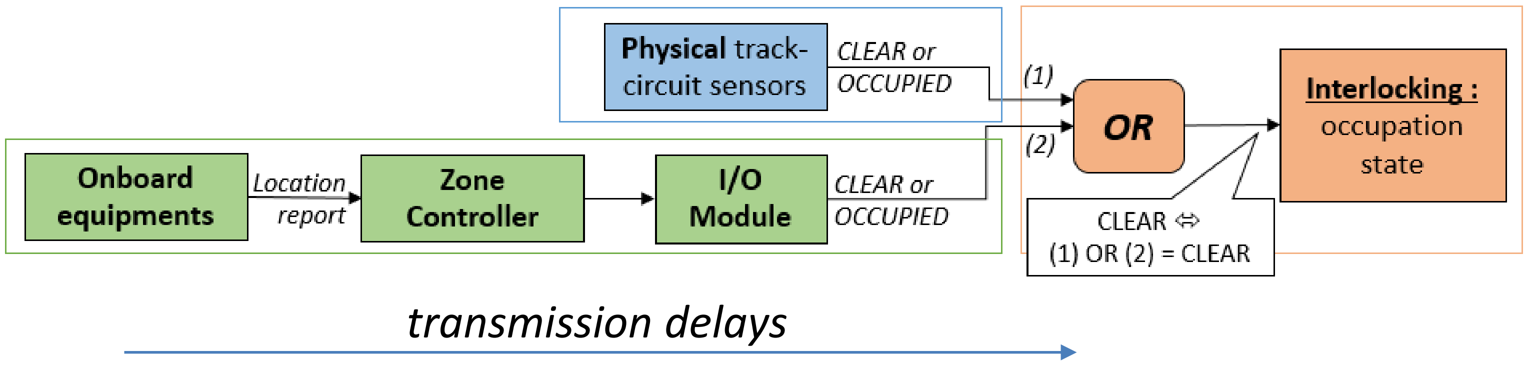
Scenario with track circuit failure



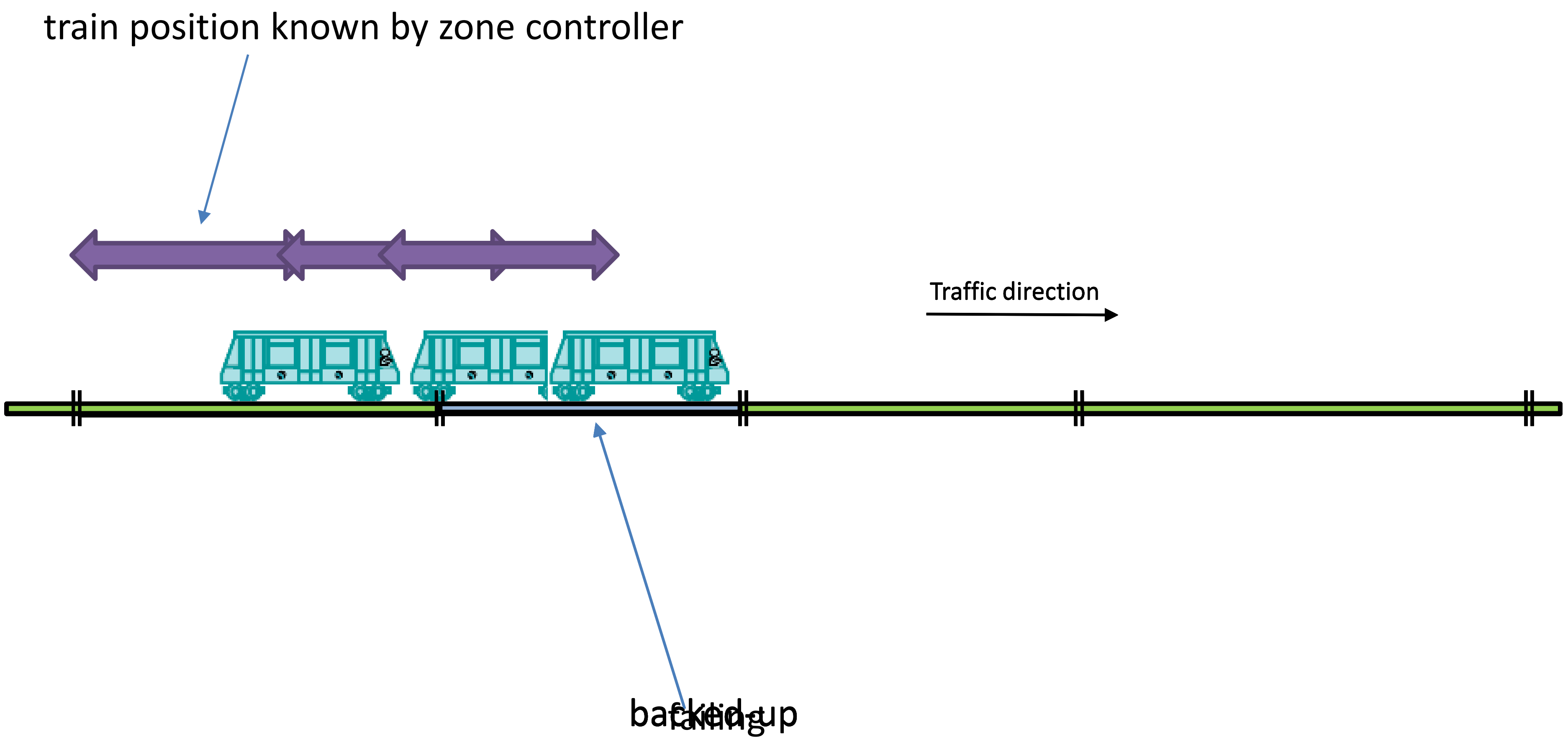
Track circuit backup: definition

- CBTC tracks equipped trains based on location messages they send.
- CBTC tracks other trains based on track circuits and other detection devices.

- In case of failure, the CBTC can supply track circuit occupation to interlocking.



- Issue: the output of this function can be delayed compared to a direct connection between the track circuit and interlocking.



≡ Contribution: identify and prove properties on track circuits that we estimate sufficient for interlocking experts to guarantee the protection zones.

Track circuit backup: definition

- Property guaranteed by the CBTC:

When a train circulates on an oriented track portion covered by a set of track circuits, there exists continuously a so-called « trailing track circuit » such that
- the output to interlocking is occupied;
- the tail of train is downstream the area covered by this track circuit.

- The argument identifies hypotheses sufficient for this property to be true:
  - trains: minimal length, maximal speed of trains
  - train localisation: precision, freshness threshold for train localisation
  - track circuits: delay for the (physical) liberation, gap of shunt
  - etc.
- Hypotheses provide equations between those parameters.
  - Reusability: parameters vary line to line.

Lessons learnt

- Scenarios play a fundamental role
  - *Exploratory* scenarios to acquire domain expertise
  - *Explanatory* scenarios to justify hypotheses

- The validation of proof mechanisms by Event-B is *fruitful* to uncover corner cases.

- We do *not* model the whole CBTC in Event-B:
  - one proof mechanism (property) at a time
  - only the aspects relevant for a given proof mechanism (property)
  - ☛ *lean models*

- We are able to extract and state a set of hypotheses sufficient to establish a given property.
  - We estimate this is the main benefit of this work
  - We identify possibly superfluous safety requirements in the input document

Lessons learnt

- Interaction with expert domains is of paramount importance.
  - The proof mechanism reflects the know-how and know-why of the system designers
  - Clarifications; validation of hypotheses

- The necessity to build a proof emphasizes the importance of some mechanisms that have greater consequences on safety than one could think… (track circuit backup).

- The safety analysis team must be seen as constructive by the design team.

- Performing safety analysis during the design phase would seem to be more effective.
  - Sharing of arguments between teams.
  - Optimization of sub-system requirements.

Prospective

- The properties of objects involved in our sufficient hypotheses can be used for *data validation* on the lines where Octys is deployed.
- Use as a *theory (trailing track circuit, protection zones…)* for future evolutions of Octys or similar products.
- Sub-system properties are requirements for sub-system suppliers.
  - Their formalization could be used as *proof goals* for the sub-systems.

Safety Analysi CBTC System: A Ri us Approach Event-B

Questions?

Mathieu Comptier, David Déharbe, Julien Molinero Perez, Louis Mussat, Thibaut Pierre, and Denis Sabatier
RSSRail – International Conference on Reliability, Safety and Security of Railway Systems: Modelling, Analysis, Verification and Certification
November 15th 2017
Pistoia, Italy

Practical

- Top-level model identifies the interplay between Octys and interlocking with respect to the safety properties
  - interlocking hypotheses are terminal: to be validated by experts
  - other hypotheses are non-terminal: require specific arguments
- Hierarchical decomposition guided by the properties the protection zones have to comply with.
- Results in a collection of safety arguments addressing different target properties.

Key points

≡ *Property-guided* analysis of the system

$\neq$ sequential inspection of the specification

≡ Properties are expressed unambiguously.

≡ Terminal hypotheses are validated by third-party experts.

≡ Non-terminal hypotheses are subject to dedicated rigorous safety arguments.

≡ Domain experts review the proof mechanism and validate it matches the actual system.

≡ Tools are used to check the proof mechanism is logically sound.

Deliverable

- Each safety argument is delivered in a dedicated Word document.
- The document addresses a *single* property.
- The document collects all the *hypotheses* needed in the safety argument for this property.
- The full *proof mechanism* establishing the safety argument is described.
- The *formalization* of the proof mechanism in Event-B is also included in the document.

Prospective

- Use of formalization by designers as a *theory of protection zones* to support future evolutions of Octys.

- Some terminal hypotheses could be target rules for *data validation* on the lines where Octys is deployed.

- Sub-system properties are requirements for sub-system suppliers.
  - Their formalization could be used as *proof goals* for the sub-systems.

Approach

- ▶ absence of collisions (front, side and rear)
- ▶ absence of derailments over uncontrolled switches

≡ Goal: prove safety properties

≡ Approach: build rigorous proof mechanisms (= arguments)

- ▶ readable by anyone familiar with the domain of application

▶ CBTC

≡ *Proof mechanism* = demonstration based on formal hypotheses

- ▶ desired property is stated clearly
- ▶ *demonstration*: logical proof expressed in natural language and checked with tools
- ▶ *hypothesis* = mathematical assumption
  - justified: presentation of a catastrophic scenario when hypothesis is not met
  - *non-terminal*: require other argument
  - *terminal*:
    - subsystem requirement found in the input documents
    - missing information, accepted after validation by expert
    - physical property