The 4SECURail Formal Methods Demonstrator

RSSRAIL 2022, June 2, Paris

Franco Mazzanti
Dimitri Belli

Italian National Research Council
Institute of Information Science and Technology – ISTI - Pisa

The railway context

The railway infrastructure is a complex **System of Systems**

Spreading across many national borders

Managed by many administrative bodies

Developed by many producers

Expensive to develop, maintain and exercise safely

The railway context

The solution: *High Quality Standard Interfaces between components*

+ to reduce costs and vendors lock-in
+ to increase competitivity, dependability and efficiency

(*safety is already guaranteed*)

Several initiatives try to advance the state of art

(e.g. EULYNX / ERTMS / SHIFT2RAIL / Europe's Rail)

recognizing the importance of formal analysis

(during development and during standardization)

4SECURail: The Demonstrator

4SECURail (November 2019 - Novemer 2021)
is a (small) project of the Shitf2Rail initiative

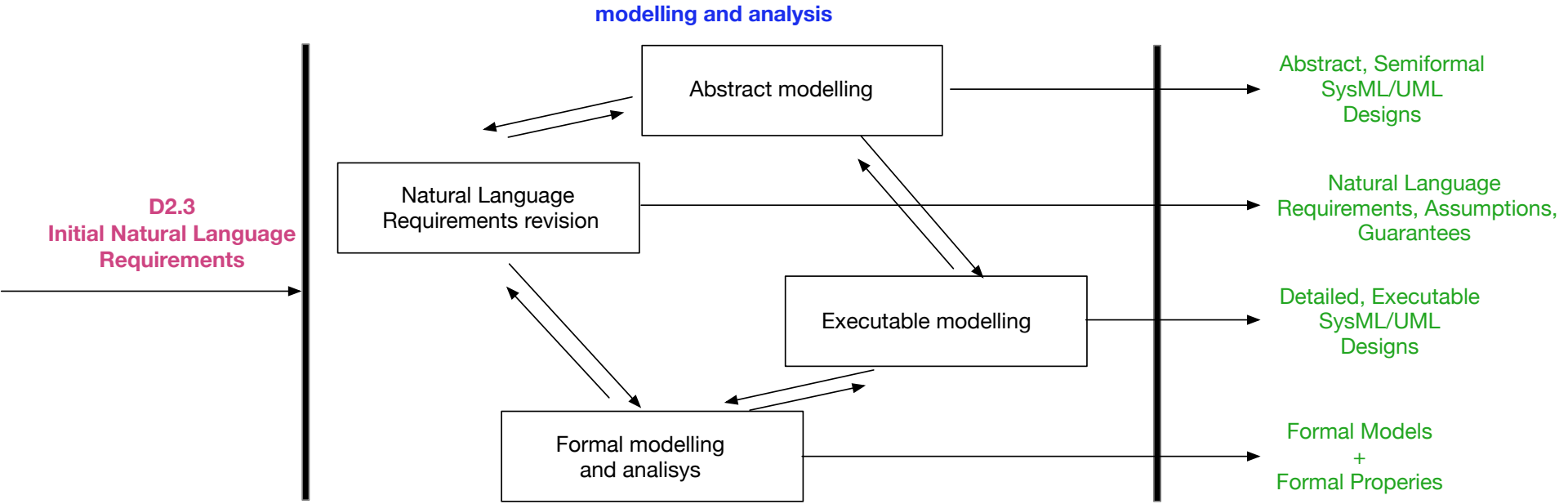
One of its golas is a **controlled experiment** (*demonstrator*) in exploiting formal methods *in the requirements definition phase* of a railway signalling system.

- Can formal methods help improving the quality of requirement specifications (standards)? ***How?***

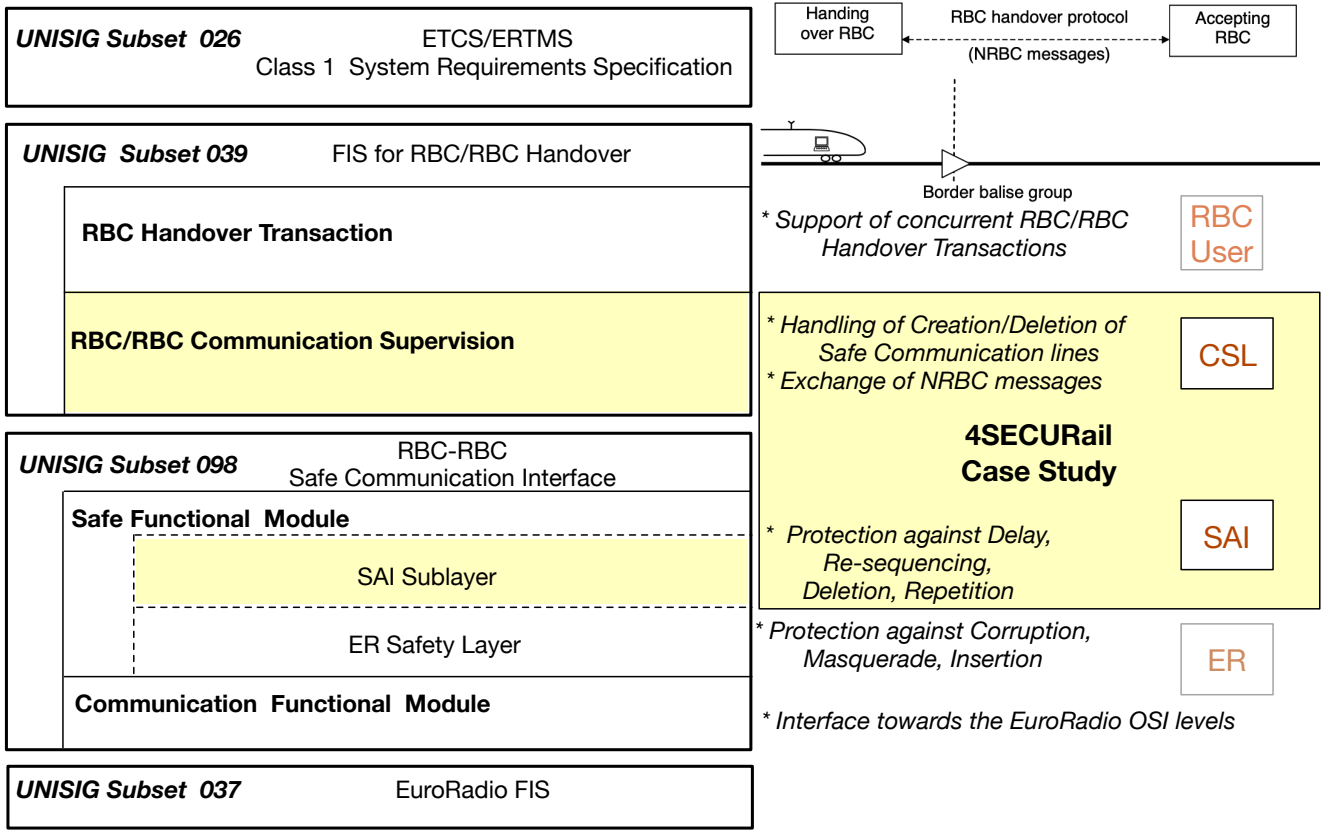
- Can their adoption be cost effective for IM? ***How much?***

I.e. The project takes the point of view of the Infrastructure Manager (standardization bodies), with focus not just in safety but also interoperabiity

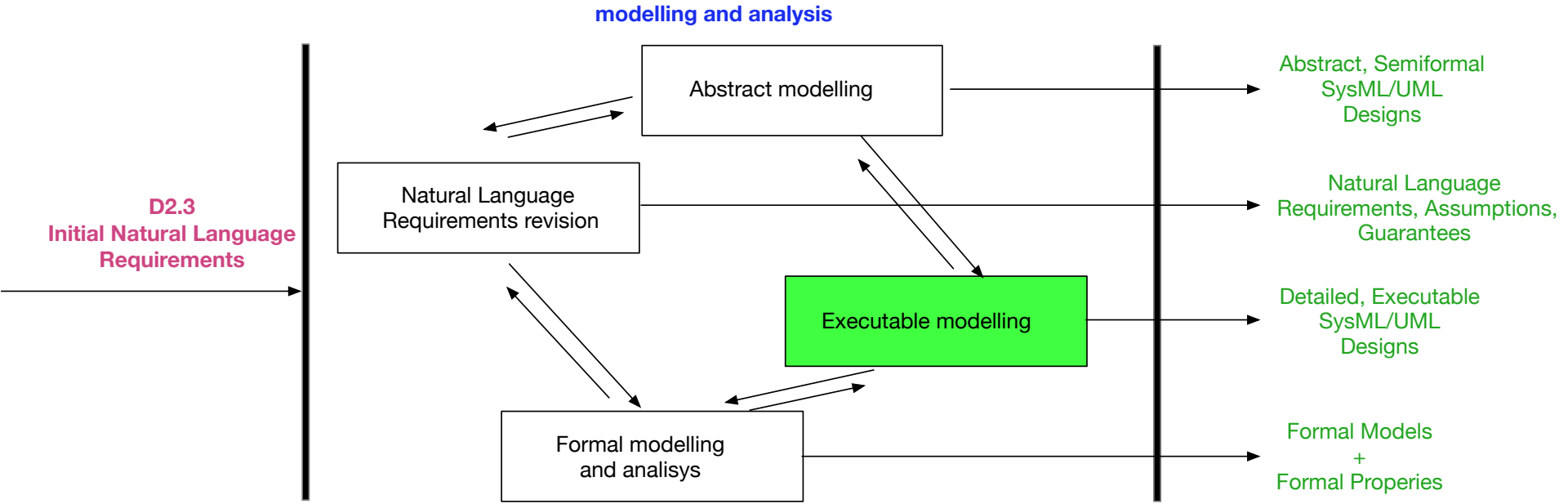
The 4SECURail approach (incremental/iterative)



The 4SECURail case study (RBC-RBC(Radio Block Centre) communications)



4SECURail: The Artifacts of the Demonstrator



Why an Executable UML/SysML model?

- Removing ambiguity in the initial NL documentation by adopting a standard, widely known, precise notation.

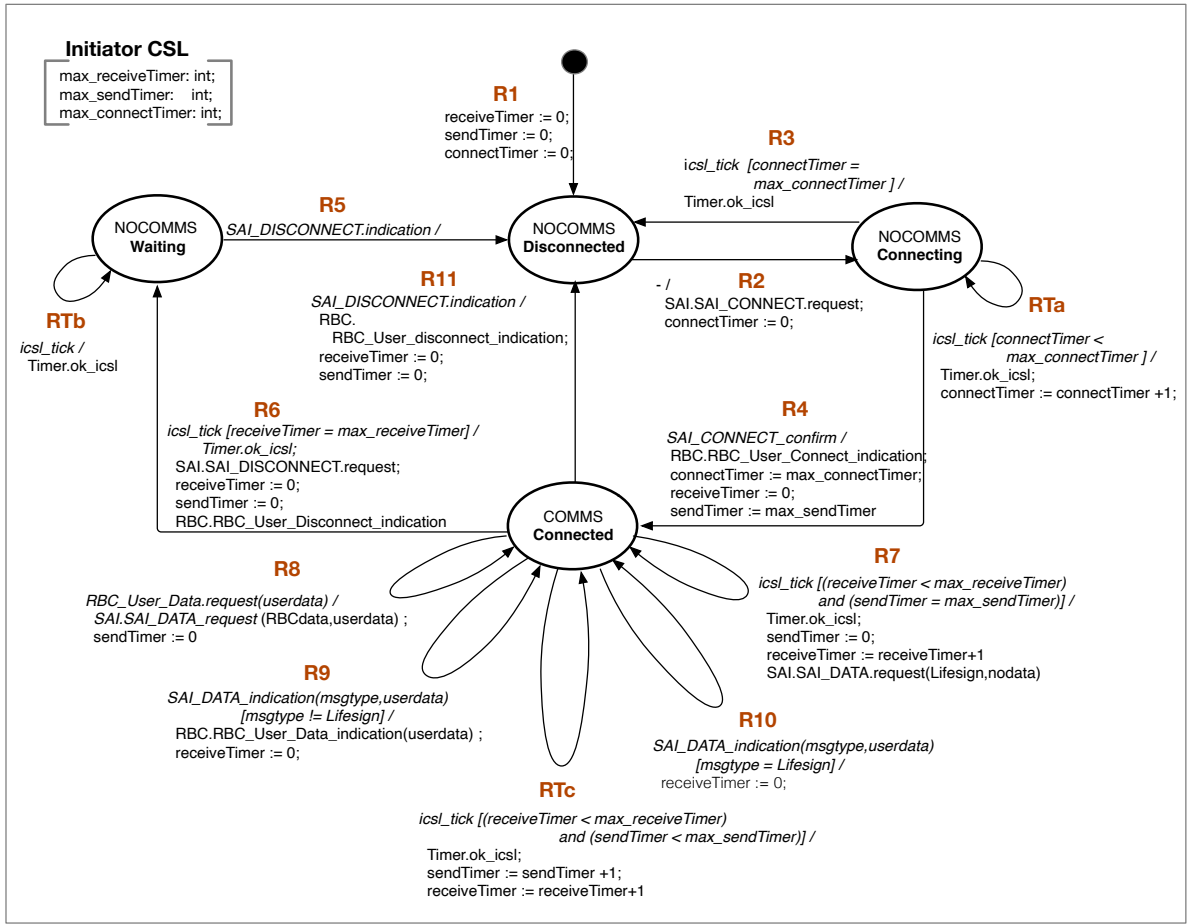
- Allowing, not formal methods experts, to understand and confirm the underlying design being modelled.

- Remaining at this level independent from the specific formal verification framework(s) adopted (preferrable in the case of international standards)

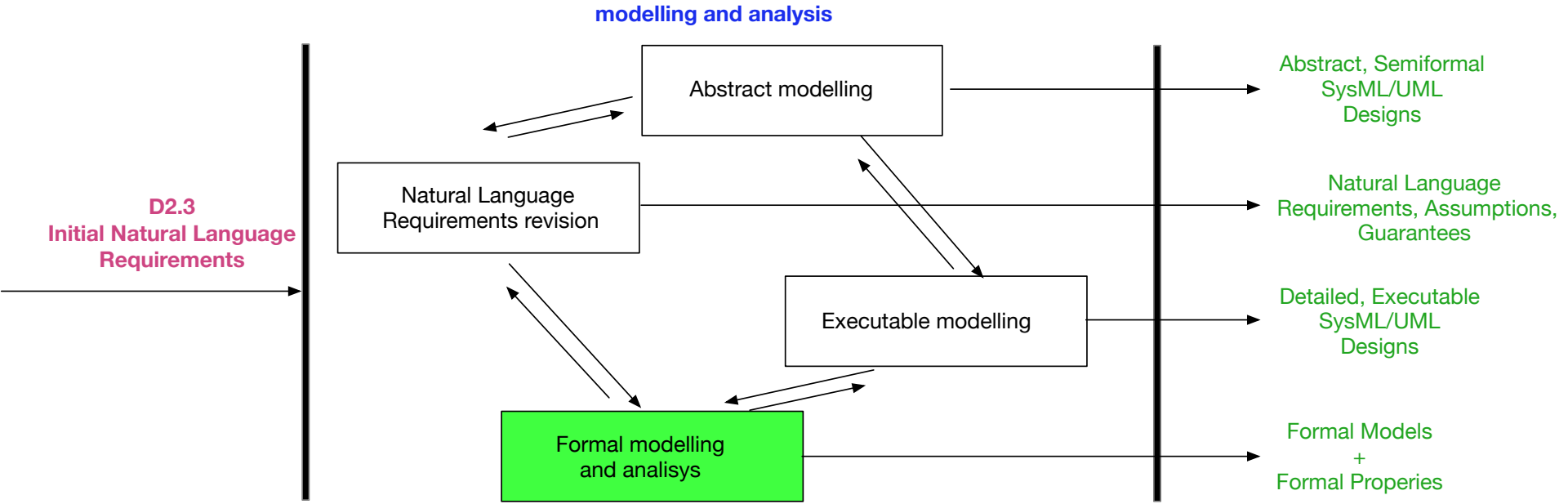
4SECURail: UML Assumptions for simple and precise semantics

- FIFO events queues
- No priority conflicts
- No parallel or composite states
- No deferred events
- No history/deep-history states
- Basic data types (enum, int, bool, vectors)
- Basic statements (assignments, conditionals)
- No entry/exit/do activities

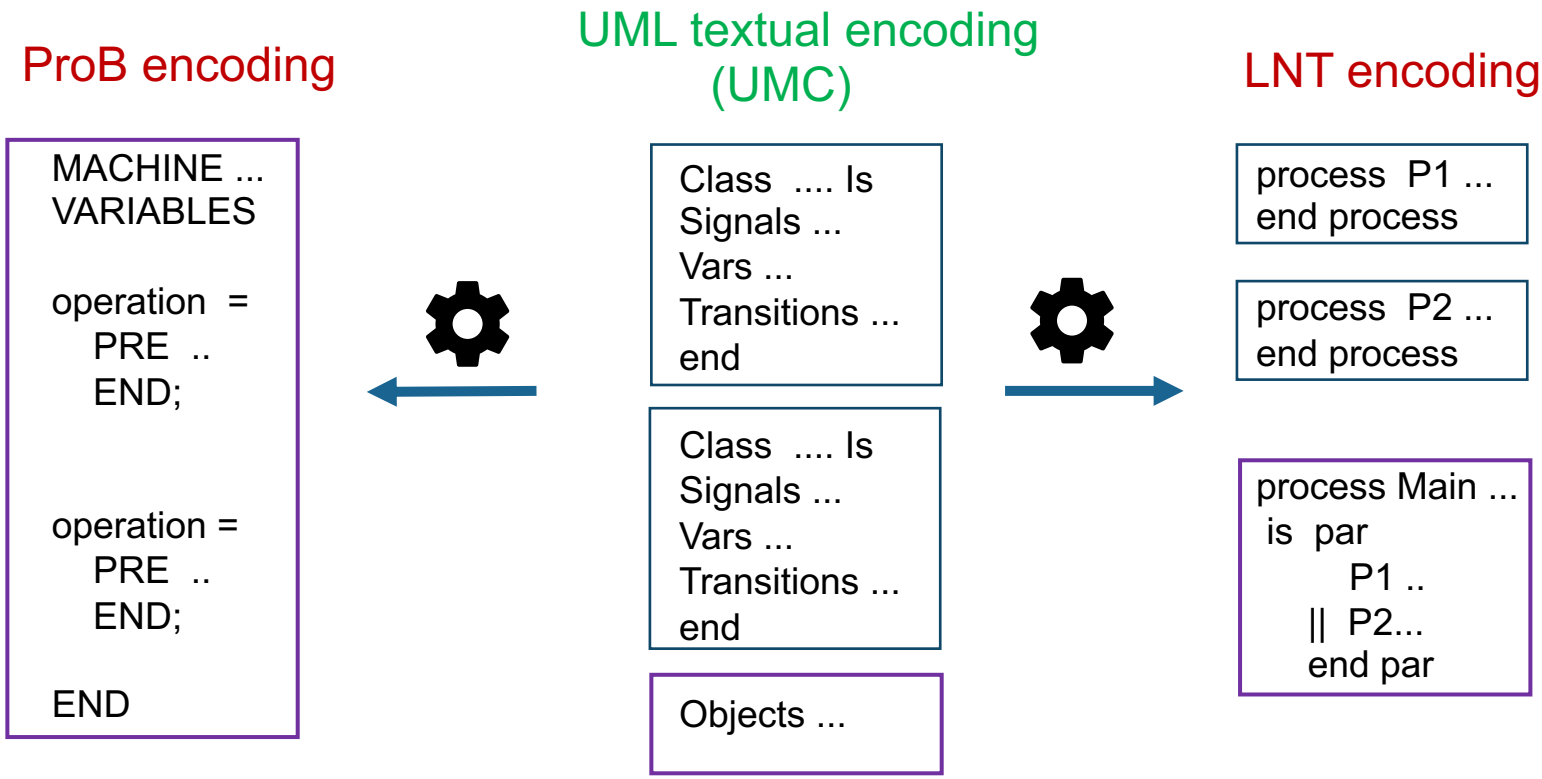
4SECURail: Executable UML Modelling (example)



4SECURail: from Executable to Formal



Formal Modelling and Analysis (1)



4SEURail: Why three formal models (UMC, ProB, LNT)?

- The three formal models can be compared for equivalence, detecting possible errors made in the formal encoding.

- The three different verification frameworks provide different verification functionalities. (e.g. linear vs branching time, compositional vs explicit)

- When the same fuctionality is supported (e.g. animation, analisys of counter examples), the most user-friendly framework can be used.

- It is however more expensive and difficult to become expert users of several verification frameworks.

4SECURail: Formal Modelling and Analysis

ProB

- Static Analysis
- Reachability Properties
- Statespace Stats
- State Invariants
- Deadlocks
- LTLe Model Checking
- CTLe Model Checking
- ...

UMC

- Static Analysis
- Reachability Properties
- System Traces Minimization
- Statespace Stats
- Deadlocks
- Runtime Errors
- UCTL Model Checking (state/event based)
- Custom system observations
- Explanations as Message Sequence Diagrams

LNT

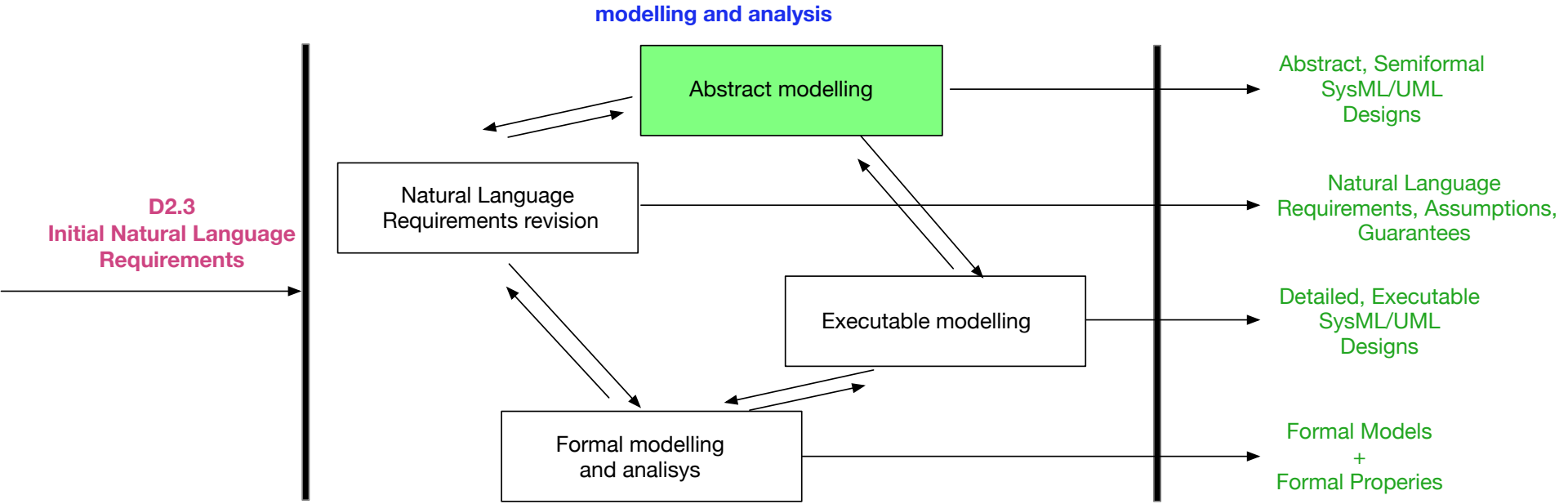
- Static Analysis
- Reachability Properties
- Statespace Stats
- Deadlocks
- MCL Model Checking (event based)
- Compositional Verification
- Strong/ Divbranching/ Sharp Minimizations
- Powerful scripting language
- ...

4SECURail: different levels of complexity of analysis

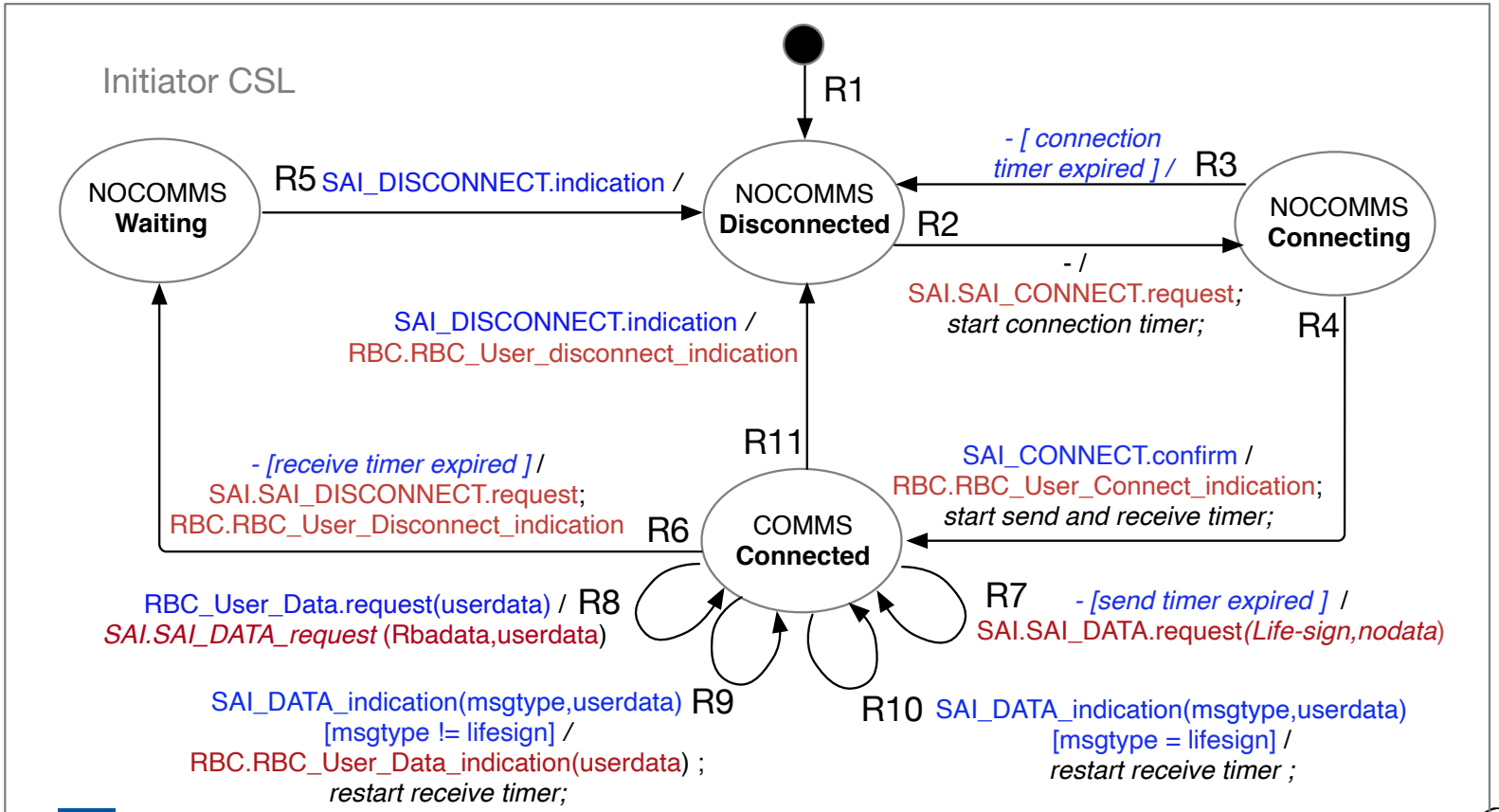
- Simple «push-button» like formal analysis
  (static analysis, reachability analysis, deadlock checking)

- More advanced verifications (model checking temporal logic formalas, compositional analysis, bisimulations and equivalences)

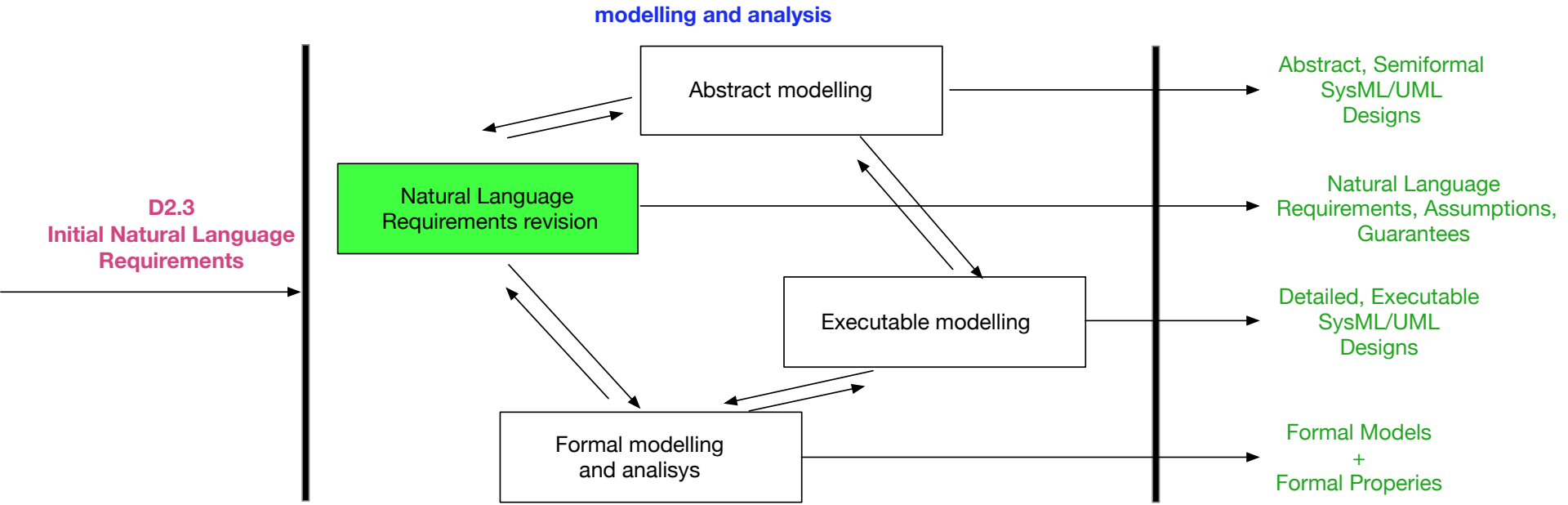
4SECURail: back from Formal Models to Natural Laguage



4SECURail: hiding non essential implementation details



4SECURail: The Approach of the Demonstrator



4SECURAIL: from Abstract Modelling to NL Requirements

- UML transitions directly mapped to NL requirements on control flow.

- Explicit definition, for ech component, of the assumptions it makes on the rest of the system, and the guarantees of which it is responsible.

- Rigorous specification of the syntactic interface between component.

Conclusions and Observations:

- The construction of the **executable model** already reveals all the NL **ambiguities**, *part* of the **inconsistences**, and **missing points**.

- Fo**rmal methods diversity** allows to detect **errors** in the formal models encoding, as well as in the translation and verification tools.

- **Formal analisys** of the executable model allows to detect **errors** in the implementation, to identify hidden **assumptions**, and to assess the expected **guarantees** of the various components.

- In the really "early" stages of requirements definition, makes sense to investigate the "reverse" flow: from Formal Models to Natural Language

4SECURail: Demonstrator References

4SECURail website: https://4securail.eu

4SECURail Deliverables doi: 10.5281/zenodo.5807738

- D2.1 Initial rationale for demonstrator structure
- D2.3 Initial case study requirements definition
- D2.5 The formal methods demonstrator experiment

Revised case study requirements doi: 10.5281/zenodo.5541217
Formal models and scenarios doi: 10.5281/zenodo.5541307
Model transformation tools doi: 10.5281/zenodo.5541350

4SECURail: Structured Natural Language Requirements

Configuration Parameters ..
External Interactions ...
External Guarantees ...
External Assumptions ...

Behavioral Requirements ...

R2: When in Disconnected state, the CSL immediately sends a SAI_CONNECT.request to the SAI component, starts a connTimer, and moves to the Connecting state.

R3: When in Connecting state the connTimer expires, the CSL moves to Disconnected state.

4SECURail

Thanks!

RSSRail 2022, Paris

Franco Mazzanti
Dimitri Belli

ISTI-CNR