



DCDS' 11

unity, solidarity, universality

## Practical formal validation method for interlocking systems

*DSCS' 11 - 15 June 2011*

**Marc ANTONI** Dr.-Ing. Ing. Supélec FIRSE  
SNCF Infrastructure Direction [marc.antonio@sncf.fr](mailto:marc.antonio@sncf.fr)

Contents

## DCDS' 11

### Contents

- Aims
- Problematic of IT critical systems
- Computerized critical module
- Formal validation method
- Formal validation tools chain
- Conclusion

State machine formalization language and Formal proof for Interlocking systems



2

# DCDS' 11

The aim of our project was to provide the Infrastructure Manager with an operating method for the formal validation of an interlocking systems.

Goal => a formal proof method by assertion, which is applicable to industrial automation equipment such as interlocking systems, and which covers equally the specification and its real software implementation.

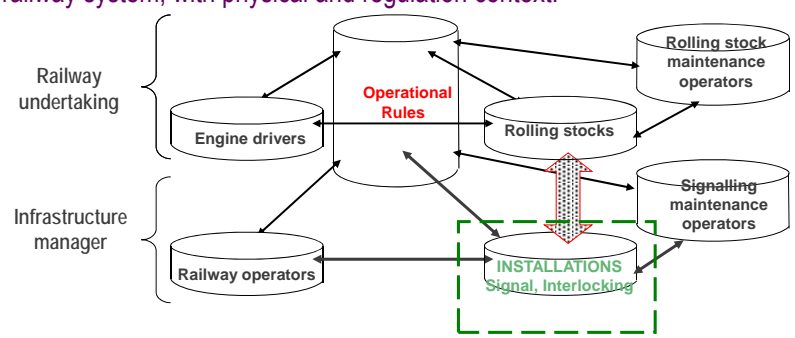
Goal of a formal method is to:

- reduce costs of all the life cycle (testing procedures, modifications...)
- be applicable to future interlocking or ERTMS systems
- increase the safety level proving that the system respects at any time all the safety properties and the postulates (that isn't the case today)



# DCDS' 11

Interlocking system safety properties have to be considered in the entire railway system, with physical and regulation context:



Postulates of the proofs => Domain of possibilities to be taken into account for the validation of safety installations before putting in service: interlocking and signaling.

## Aims of the project

# DCDS' 11

---

Interlocking system safety properties have to be considered in the entire railway system, with physical and regulation context:

ESW Critical computerized system Over system

Functional Software

Hard and Ground Software

Operators Maintenance

sensors

Exploitation rules ERTMS system

Rolling stocks Field Elements

Block system

*infra*

5

## Contents

# DCDS' 11

---

### Contents

- Aims
- Problematic of IT critical systems
- Computerized critical module
- Formal validation method
- Formal validation tools chain
- Conclusion

*infra*

6

## Problematic of IT-Systems systems

## DCDS' 11

Many recent experiences show us that the current development methods don't give a "real guarantee" that the products are safe, that they can be integrated safely in a global railway system.

- A recent study showed that more than  $\frac{3}{4}$  accidents in relation with computerized systems are due to specifications errors
- Examples are numerous, inside and outside of the railway domain (in particular then the IT systems are complex and don't take into account the overall system...)
- The current standards are not sufficient: "process method obligation" and not "result obligation" in terms of security, safety and availability.



7

## Problematic of IT-Systems systems

## DCDS' 11

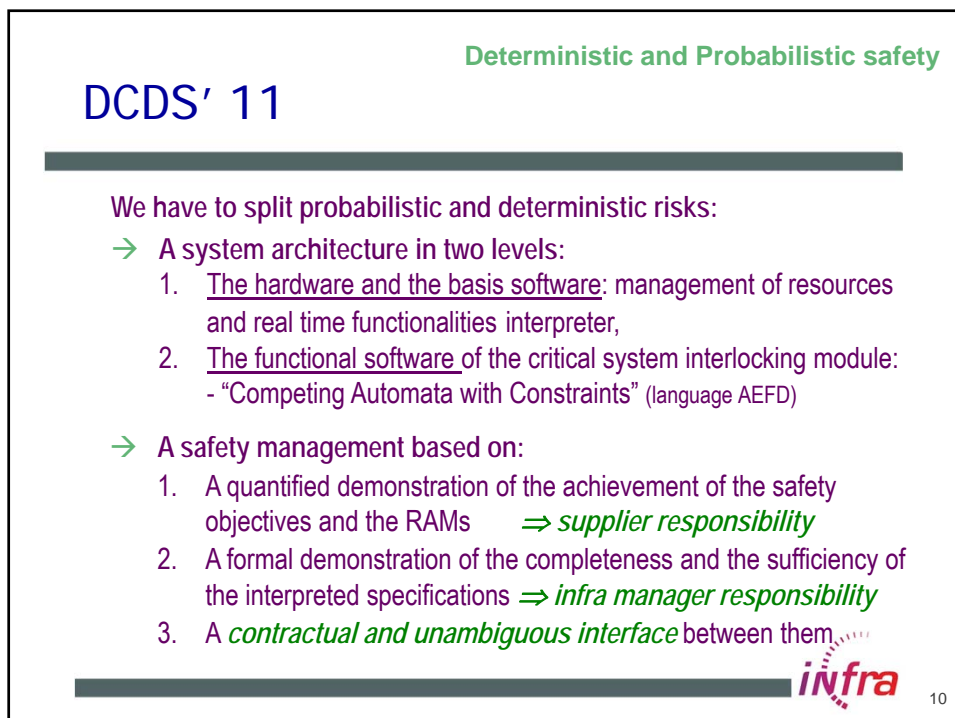
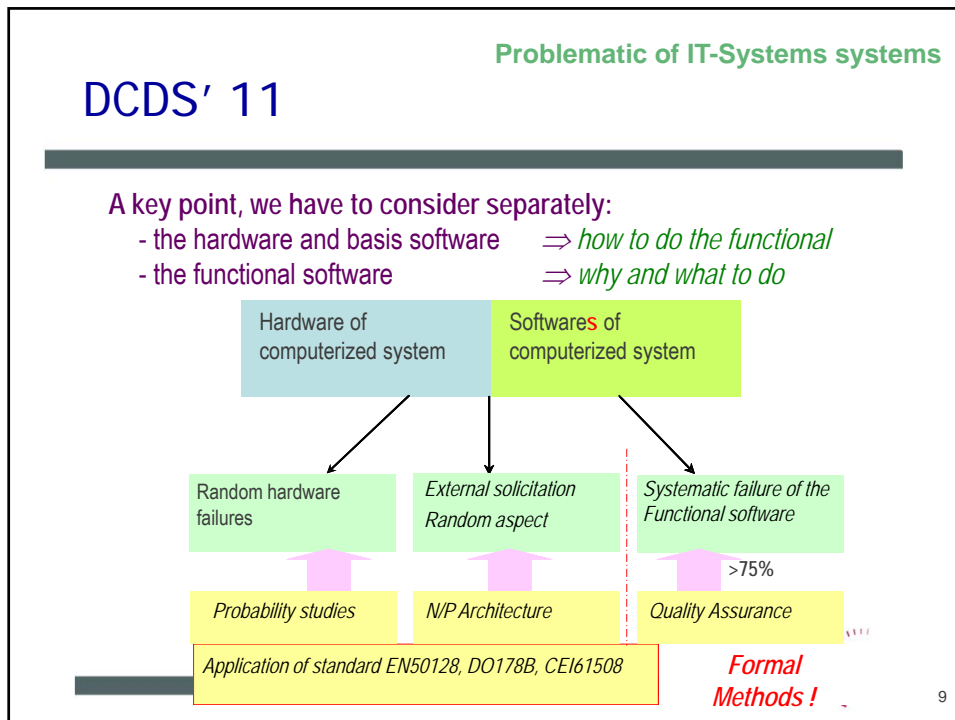
We need a new way to give a real safety guarantee of critical computerized systems:

- With computerized systems:
  - the list of the dreaded events is not countable,
  - it is necessary to define the frame of the authorized system states and to be able to check the framework is never left,
  - a formal validation proof is only possible if the domain of the reachable system states is finished,
  - we have to distinguish formal validation and formal verification.

⇒ *an application designed with an algorithmic software is generally impossible to prove*



8



Contents


## DCDS' 11

---

Contents

- Aims
- Problematic of IT critical systems
- **Computerized critical module**
- Formal validation method
- Formal validation tools chain
- Conclusion

---


11

Computerized critical module


## DCDS' 11

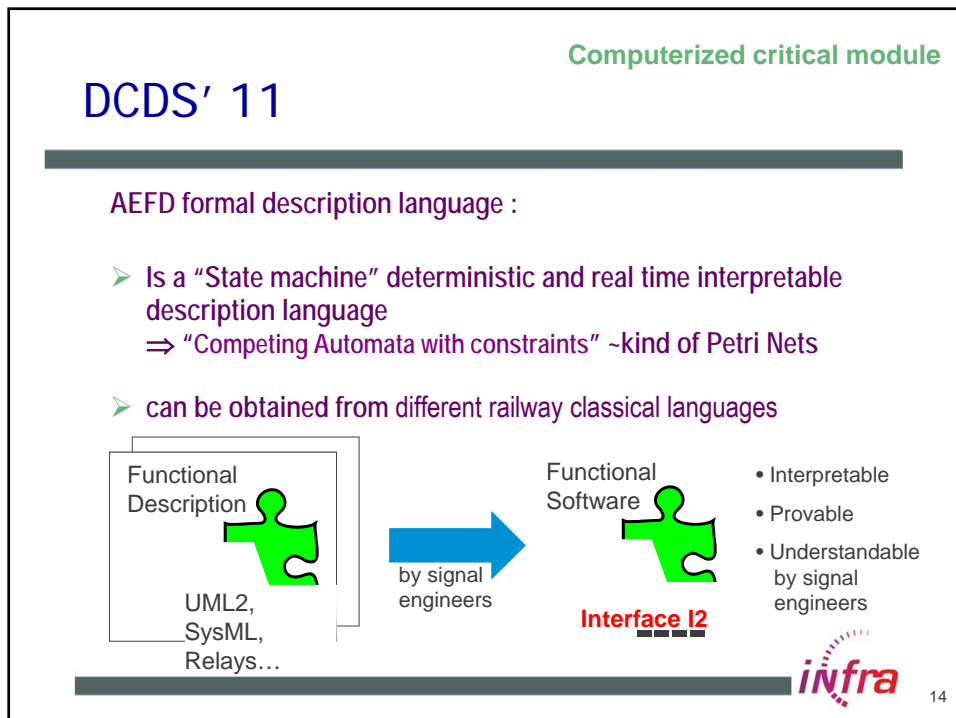
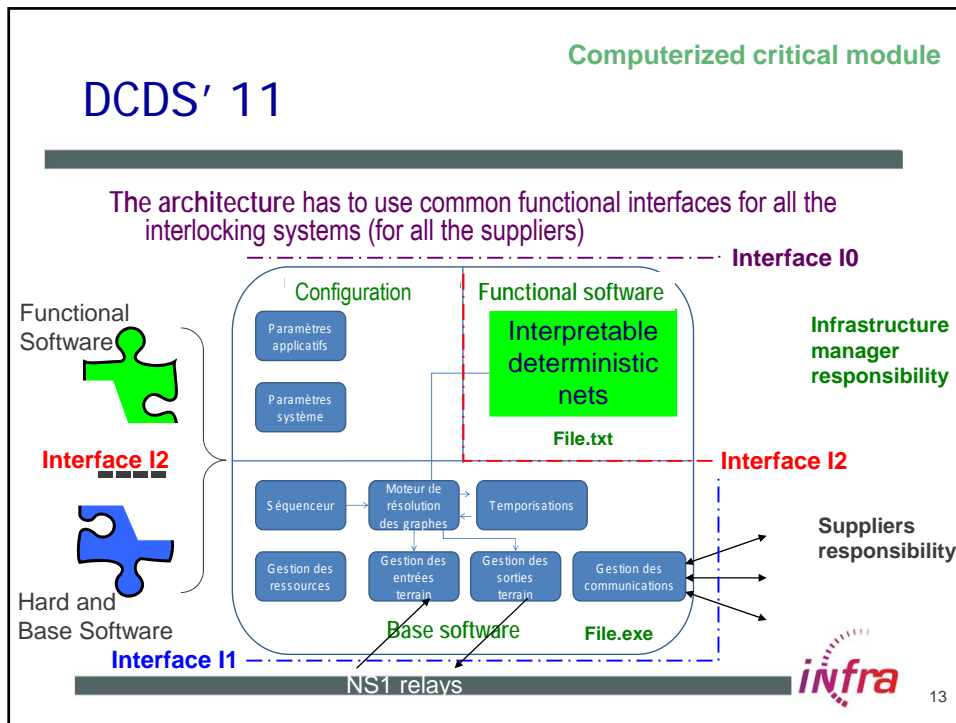
---

The new ESW critical interlocking system could be able to carry out :

- A clear separation between « hardware & basic software » (suppliers view) and « functional software » (infrastructure manager view),
- Clear interfaces between the computerized module and rest of the railway system,
- Specification & functional software with interpretable deterministic Petri nets (interpreted in the target machine),
- A formal validation of the functional software in the real environment conditions of the interlocking system.

---

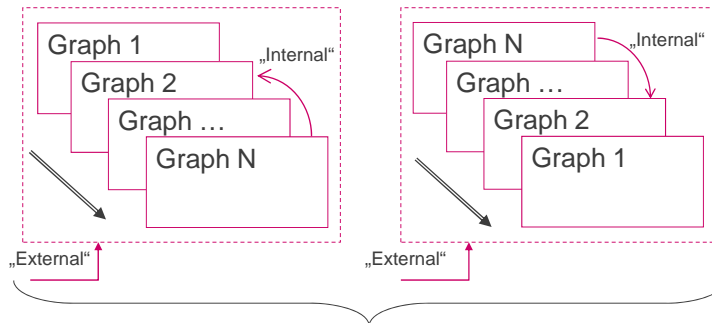

12



# DCDS' 11

Petri nets are in general not interpretable in a determinist way

- There is not differentiation between "internal " and "external" events
- There are possible indecisions in the interpretation (priorities)



Two different interpretations / Two different tree of reachable system states

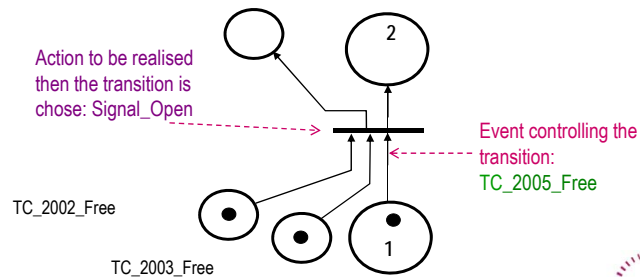


# DCDS' 11

With classical Petri networks:

- Interpretation depends on the order of graphs
- They are in general not interpretable in real time

Classical PN





## Computerized critical module

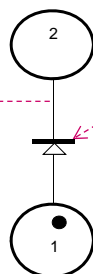
## DCDS' 11

AEFD language allows a deterministic interpretation of the signalling functions (with competitive networks with constraints):

- Interpretation is accomplishable without indecision
- Interpretation isn't dependent of the reading order of graphs
- Interpretation is accomplishable in real time

AEFD PN

Action to be realised then the transition is chose:  
Signal\_Open



Event controlling the transition:

Zone\_2005\_Free

Conditions: TC\_2002\_Free AND TC\_2003\_Free



17

## Computerized critical module

## DCDS' 11

AEFD language allows a deterministic interpretation of the signalling functions (with competitive networks with constraints):

- Interpretation is accomplishable without indecision
- Interpretation isn't dependent of the reading order of graphs
- Interpretation is accomplishable in real time

AEFD notation:  
under  
interpretable  
text file form

```

...
Nom du graphe
1
2
TC_2005_Free Event
TC_2002_Free AND TC_2003_Free AND
TC_2005_Free Condition
Signal_Open; Action
...

```

```

TC_2005_Free
[ TC_2002_Free AND
TC_2003_Libre AND
TC_2005_Free ]
Signal_Open/

```



18

**Computerized critical module**

## DCDS' 11

---

**Graph Example :**

Place origin	Place destin.	Event	Conditions	Actions
1	2	TC_2005_Free	TC_2002_Free ET TC_2003_Free	Signal_Open

*infra* 19

**Computerized critical module**

## DCDS' 11

---

**Graph Example :** description of a signaling function with places (states) and transitions (events, condition, action) ; The current state is defined by a mark (one mark by graph)

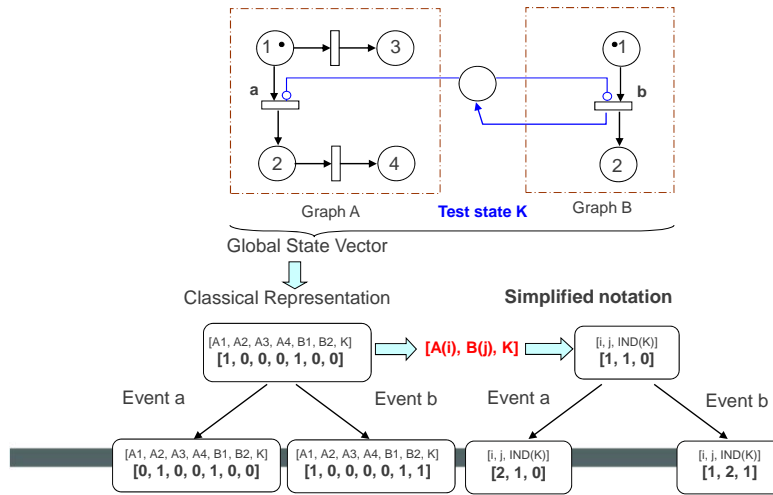
Example ; control of a switch

*infra* 20

# DCDS' 11

## Computerized critical module

- Communication between Petri nets classical graphs

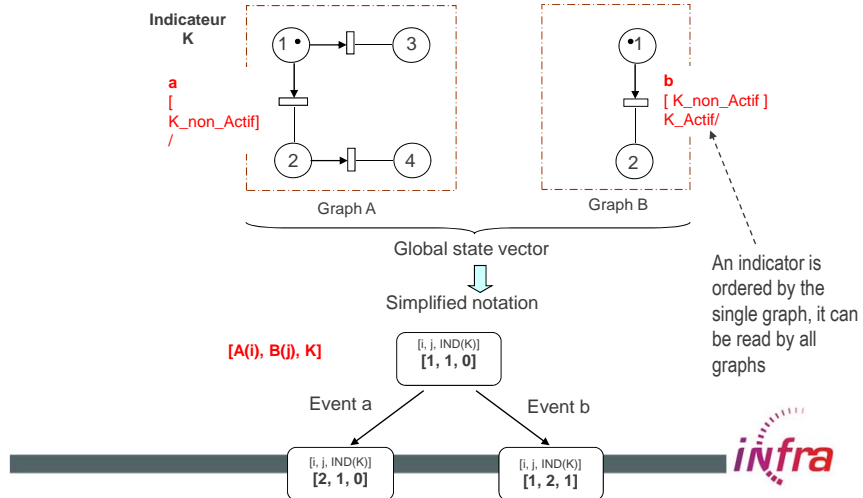


21

# DCDS' 11

## Computerized critical module

- Communication between graphs in the chosen notation (AEFD):



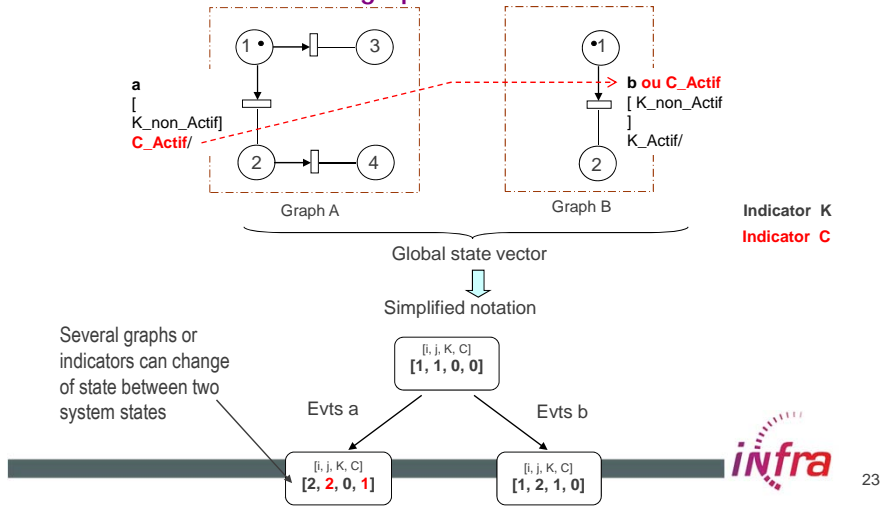
An indicator is ordered by the single graph, it can be read by all graphs



22

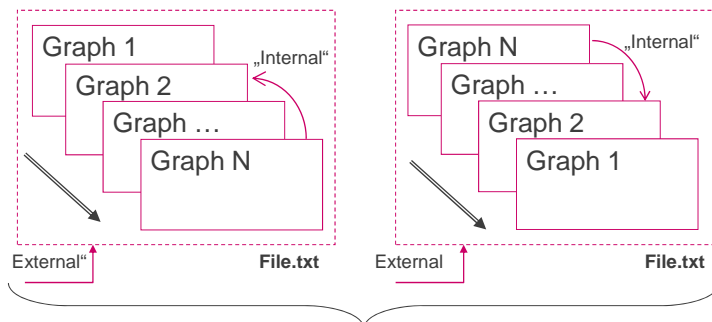
DCDS' 11

Communication between graphs in the AEFD notation:

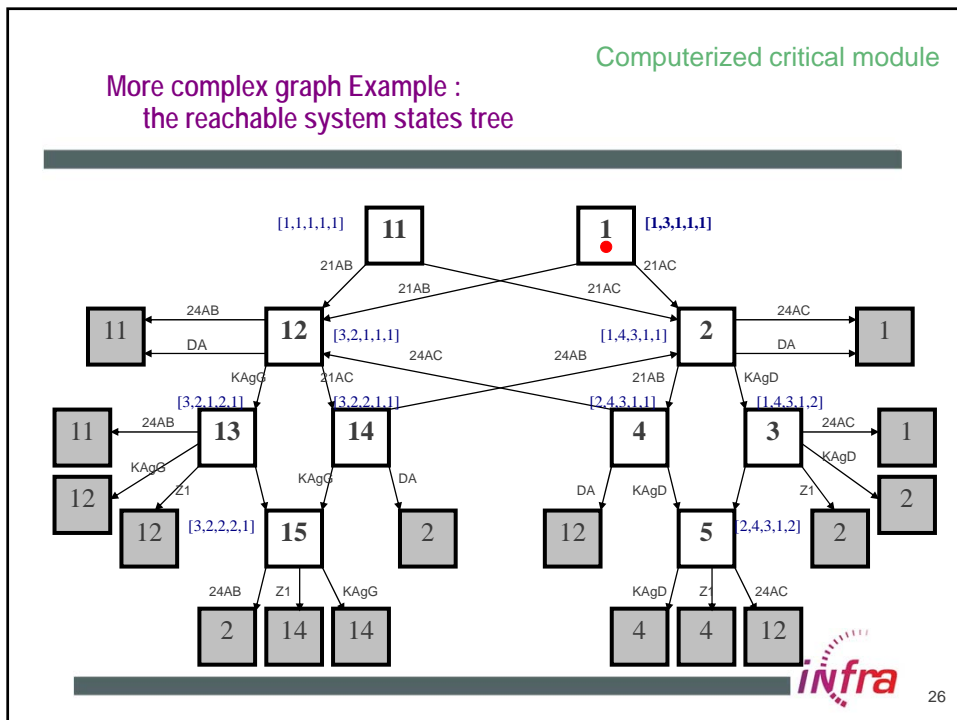
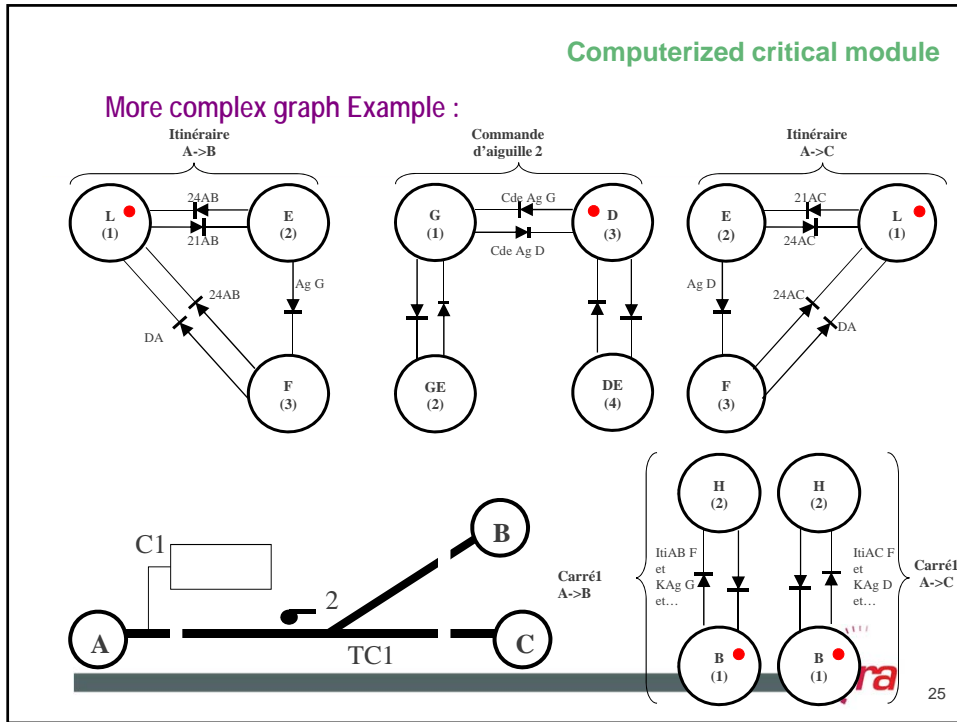


DCDS' 11

→ With the writing mode, the Petri nets are interpretable in a determinist way, without ambiguity



A unique group of reachable system states, finished and countable




Contents

## DCDS' 11

---

Contents

- Aims
- Problematic of IT critical systems
- Computerized critical module
- **Formal validation method**
- Formal validation tools chain
- Conclusion


27

Formal verification or/and validation method

## DCDS' 11

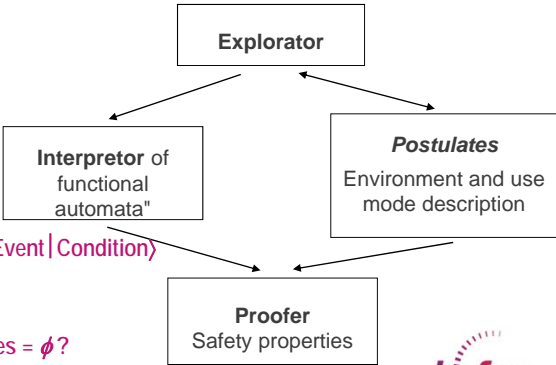
---

**Formal Validation Method:**  
 The proof is brought on the final interpreted functional model written with **Competing Automata with constraints**

Exploration of a finite state automata


VE[(j)<sup>th</sup> injection]  
 ← VE[(j-1)<sup>th</sup> injection] → T(Event | Condition)

Post\* (VE(0)) ∩ Unsafe States = ∅?



```

            graph TD
              Explorator[Explorator] --> Interpreter["Interpreter of functional automata"]
              Explorator --> Postulates["Postulates  
Environment and use mode description"]
              Interpreter --> Proofer["Proofer  
Safety properties"]
              Postulates --> Proofer
          
```

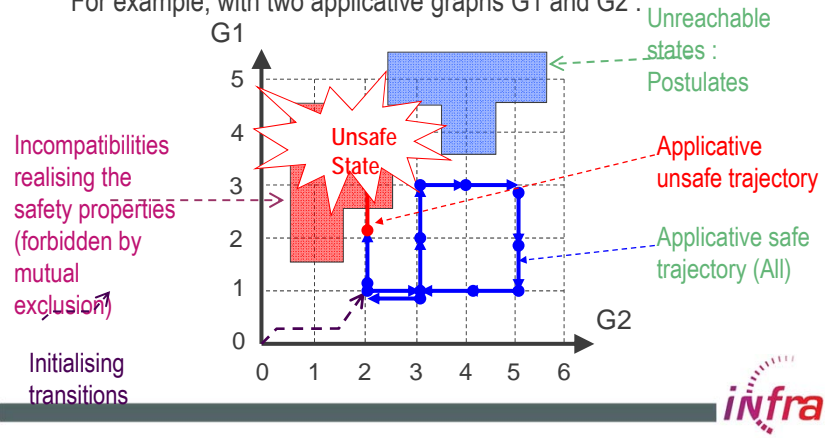

28

Formal verification or/and validation method

DCDS' 11

Formal validation has to guarantee the safety for all the possible inputs sequences: in nominal and in degraded modes.

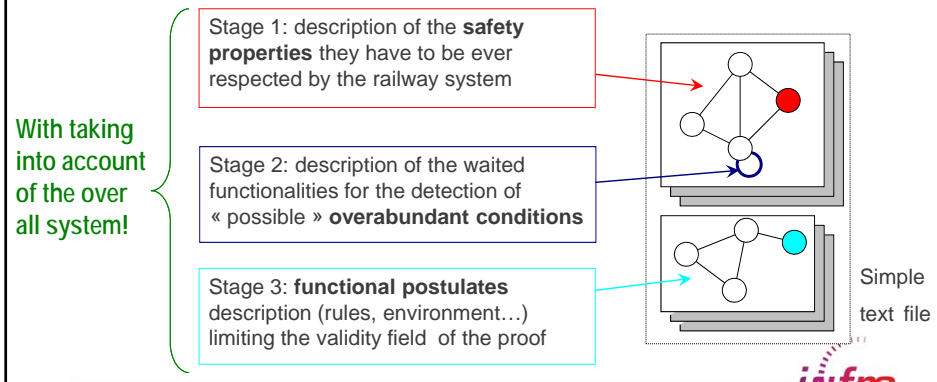
For example, with two applicative graphs G1 and G2 :



Formal verification or/and validation method

DCDS' 11

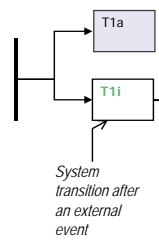
The safety properties have to be written by signalling engineers with the knowledge of signalling functions (with the help of "modelling" people)



## Formal verification or/and validation method

## DCDS' 11

All safety properties and all superfluous conditions are checked by the embedded functional application during one exhaustive exploration of the reachable states of



The initial system state is sure + All transitions are generated + All transitions are proved  $\Rightarrow$  All states are safe



31

## Contents

## DCDS' 11

## Contents

- Aims of the project
- Problematic of IT critical systems
- Deterministic and Probabilistic safety
- Computerized critical module
- Formal validation method
- **Formal validation tools chain**
- Conclusion



32



Formal validation tools chain

DCDS' 11

Tools could be developed in the future INESS context to accomplish:

- Automatic definition of the safety properties and the postulates describing the conditions of use,
- Formal writing of these properties in order to make the proof,
- Definition of the initial system state in which all the safety properties are true,
- Evaluation of the safety properties by recurrence for each transition between system states. The safety properties are evaluated until all safety properties are true, otherwise the proof is stopped.

⇒ Their application is possible by persons without special mathematical education but only a good signalling knowledge

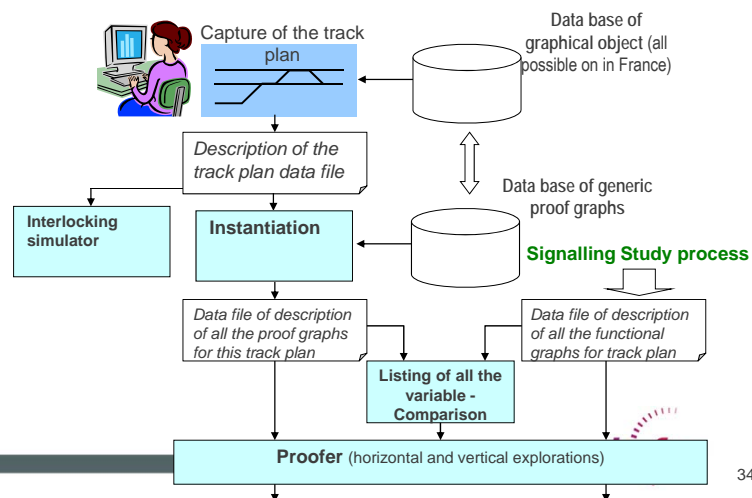
⇒ Their application leads to a significant reduction of the validation costs and delays .

33

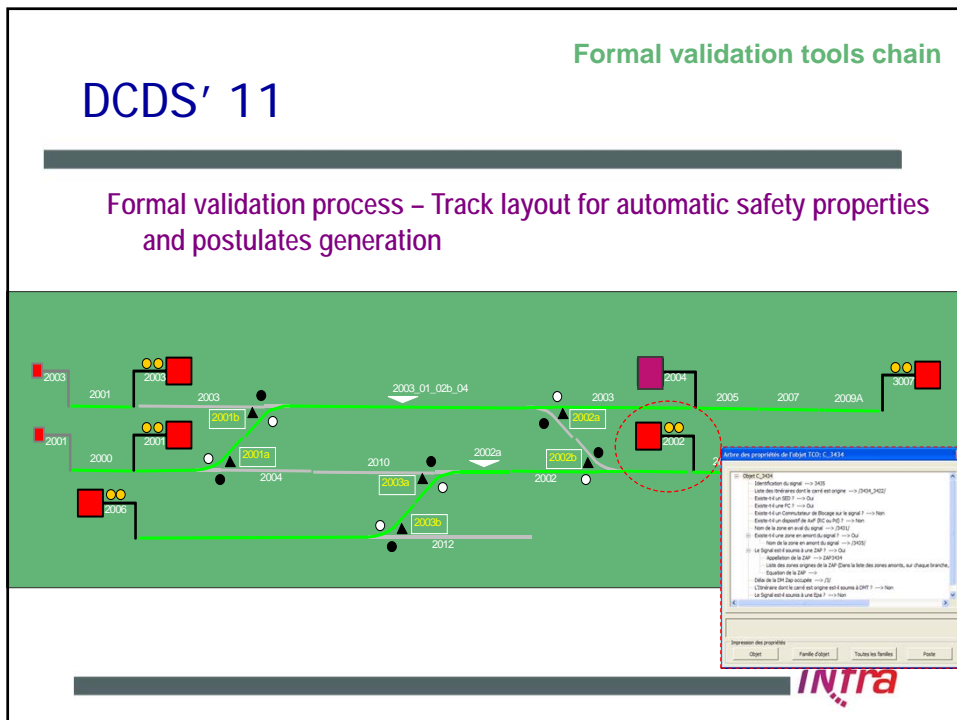
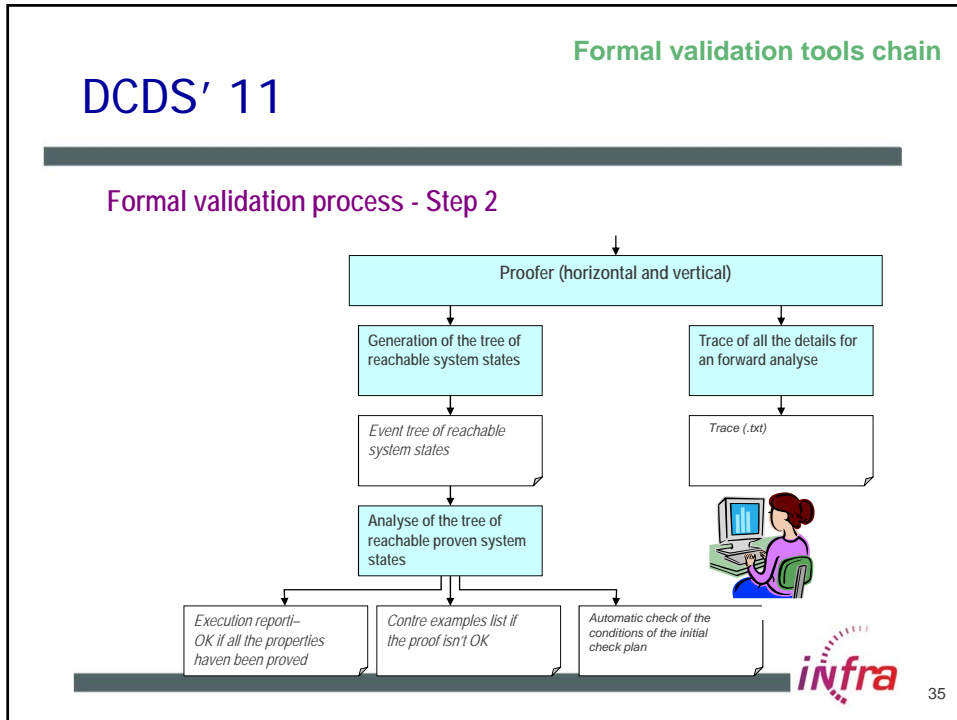
Formal validation tools chain

DCDS' 11

Formal validation process - Step 1



34



Formal validation tools chain

DCDS' 11

Formal validation process – Proof tool

Automata with marked place  
Variables  
Exploration of all potential even –  
System vector state before and after the external event

- (1) Emulation of the interpretation rules of the target machine
- (2) Exploration of all the reachable states (two algorithms)
- (3) Evaluation of all the safety properties after each new external event
- (4) Generation of complete execution trace



Formal validation tools chain

DCDS' 11

Formal validation process – Reachable system states tree

Tree of proved transitions and reachable states :

- Yellow : Postulate non respected;
- Blanc : Transition true and proven
- Grey : Transition not authorised (constraints)
- Red : Transition with un respected safety property
- Green : Transition with an over abundant condition

- (1) Vivacity check
- (2) Execution report
- (3) Presentation of the results
- (4) Elaboration of the transition tree



## DCDS' 11

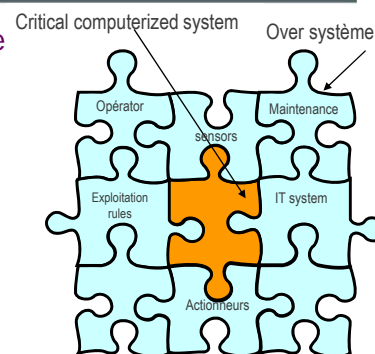
### Contents

- Aims of the project
- Problematic of IT critical systems
- Computerized critical module
- Formal validation method
- Formal validation tools chain
- Conclusion

## DCDS' 11

The proposed method allows to realize industrially a formal validation of the IT system functionalities in its context of use:

- allows an automatic and exhaustive check-up of an interlocking system,
- gives as result an achieved guaranty.



The mathematic properties of a "state machine" can be used when the interlocking system design with the necessary constraints.

**Conclusion**

## DCDS' 11

---


The approach can be a bridge between two worlds: railway vs. university

- to conceal the mathematical aspects,
- to have a interface specific to the domain.

The method allows to reduce the costs and increases the safety of critical IT system ⇒ UIC recommendation in the INESS project

**NB :** *The application of formal methods will be soon an obligation for the development of new railway critical IT system if the domain want really:*

- a safe railway world for tomorrow,
- to save people and money,
- to react before a next railway informatics Titanic,
- to maintain the safety level has an important advantage of the railway system in a competitive market.

 41

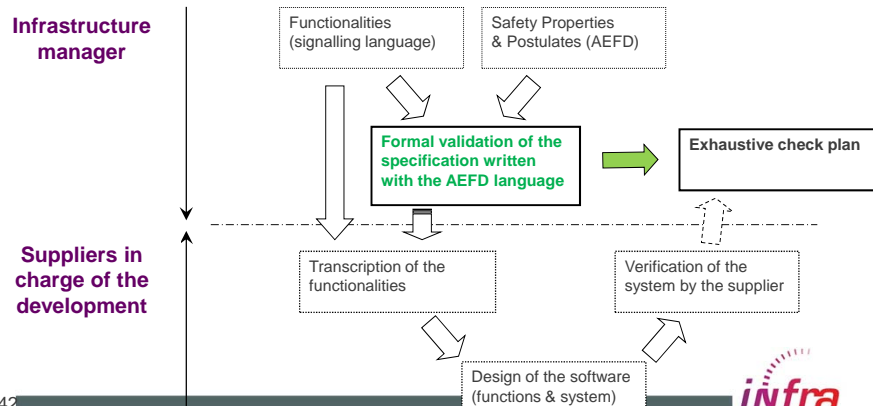
**Conclusion**

## DCDS' 11

---


**Two possible use of the formal specification for “call for tender” :**

- 1<sup>st</sup> use : proved specifications + exhaustive check plan generation



**Infrastructure manager**

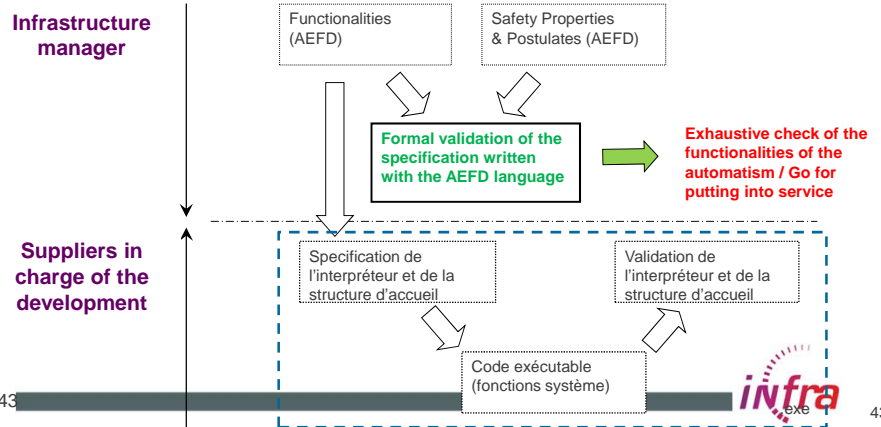
**Suppliers in charge of the development**

42  42

Conclusion

# DCDS' 11

Two possible use of the formal specification for "call for tender" :  
- 2<sup>nd</sup> use : proved specifications + interpretation by a safe target unit



43

43

## Signalling architecture for railway infrastructure System integration point of view

Engineering system can help for:

Writing functional specification in coherence with the overall system + Producing automatically documents for the both signalling and software teams + Building a tests strategy in coherence with the new interfaces.



44



### Signaling architecture for railway infrastructure System integration point of view

Engineering system can help for:

Writing functional specification in coherence with the overall system + Producing automatically documents for the both signalling and software teams + Building a tests strategy in coherence with the new interfaces.



### Signalling architecture for railway infrastructure System integration point of view

Incoherence without an entity in charge of the integration system:

1) Costs, 2) Delays, 3) Unsafe events



## Signalling architecture for railway infrastructure System integration point of view

For mastering costs and the coherence on the total life cycle of the railway:

- 1/ Responsibilities
- 2/ Context of one given overall system (regulation corpus)
- 3/ Safety properties of a new system



DCDS' 11



■ ■ ■ Thank you for your kind attention

*Marc ANTONI Dr.-Ing. FIRSE*

+33 6 29 91 77 43  
[marc.antoni@sncf.fr](mailto:marc.antoni@sncf.fr)



### Signaling architecture for railway infrastructure System integration point of view

## DCDS' 11

The particularity of these functions is all at once to define standard for some functions and to offer adaptable interfaces means with the existing installations, but also to build a part of the performances on digital transmissions.

49

### Signaling architecture for railway infrastructure System integration point of view

The system has to be considered as « white box » for the functions - Otherwise the technology (*“know-how”*) becomes more important than the functions and the interfaces with the overall system (*“know-why” and “know-what”*).

50

