FORMAL VALIDATION METHOD AND TOOLS FOR COMPUTERIZED INTERLOCKING SYSTEM

FM 2012 Industry day

Dr Marc ANTONI, SNCF,
Technological Innovation Department, France

Summary

Safety problems of IT-Systems

Railway characteristics

Interpretable deterministic Petri nets

Formal validation method

Application

Conclusion

Summary

Safety problems of IT-Systems

Railway characteristics

Interpretable deterministic Petri nets

Formal validation method

Application

Conclusion

Problematic of IT-Systems systems

The railway system uses more and more data processing or computerized systems:

The classical IT-Systems have some advantages:

- → News functions, increasingly complex
- → Orders at distances
- → Exploitation staff reduction…

They have also disadvantages – They are:

- → are longer to develop and to modify
- → are less available and have à shorter life time
- → require a qualified maintenance staff
- → are more difficult to validate and to integrate in a global system

Problematic of IT-Systems systems

The recent experience show us unfortunately that the current development methods don't give a "real guarantee" that the products will be absolutely safe (SIL4 or not), that they can be integrated safely in a global railway system.

- A recent study showed that more then ¾ accidents in relation with computerized systems are due to specifications errors
- The accidents are due to incorrect functional descriptions, to modification or maintenance operation
- The examples are numerous, also in the railway applications (cf. ETCS and ERTMS applications…)
- A fact is sure, the current standards are not sufficient… There are SIL4 and SIL4 systems…

Problematic of IT-Systems systems

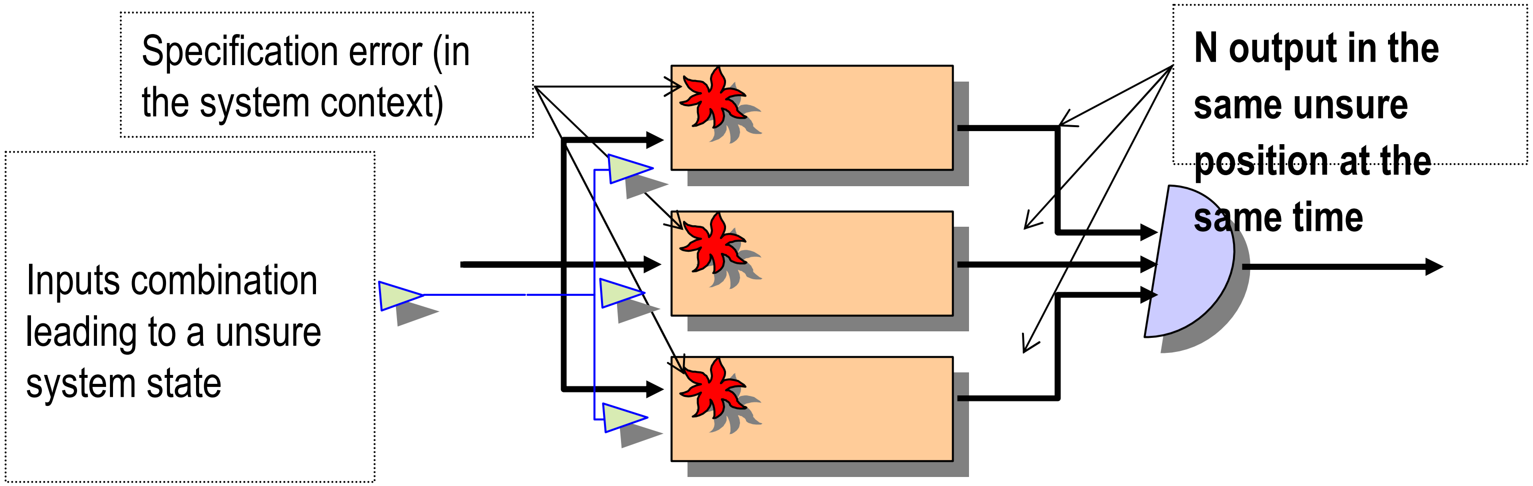
We need a new way to guarantee the safety of critical computerized systems:

→ With the traditional systems:
- it was necessary to identify the dreaded events and to reduce their probability

→ With computerized systems:
- the list of the dreaded events is not countable
- it is necessary to define the framework of the authorized system states and to be able to check the framework is never left
- an formal proof is only possible if the domain of the reachable system states is finished and countable.
- the formal proof of an application designed with an algorithmic software is „difficult ", or generally impossible to realise

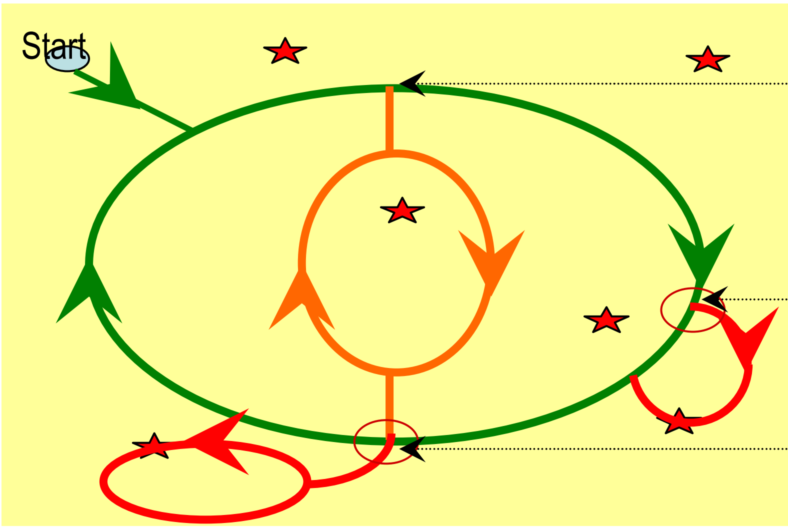
Problematic of IT-Systems systems

An N by P architecture does not reduce this kind of risk (failure)
If there exists a combination of entries which can lead the system to a unsure state, this one will exist on all the computerized units at the same time



Problematic of IT-Systems systems

A countable reachable system states is necessary to the realisation of a formal proof: If not, the system is in practice not testable…



When a envisaged combination occurs, the system runs over the greens and oranges system states, usually well tested

When a non envisaged combination occurs, the system can reach the reds system states, not tested and potentially dangerous

A formal method has to prove that it doesn't exist any not envisaged combination who can activate a unsure function

Summary

Safety problems of IT-Systems

Railway characteristics

Interpretable deterministic Petri nets

Formal validation method

Application

Conclusion

Railway characteristics

→ In general:

- The railway systems generally use Boolean values, use automatisms
- The safety is carried out with incompatibilities (exclusion in space and time of a common position of resources)

→ Interlocking functions has to:

- Take into account all the national laws, exploitation rules…
- Take into account the environment of the system (without exportation of safety constraint…)
- Be in service 24:00 over 24:00, 365 days par year, many years long…
- Are numerous on the network
- Be checked at 100% after each functional modification or maintenance intervention

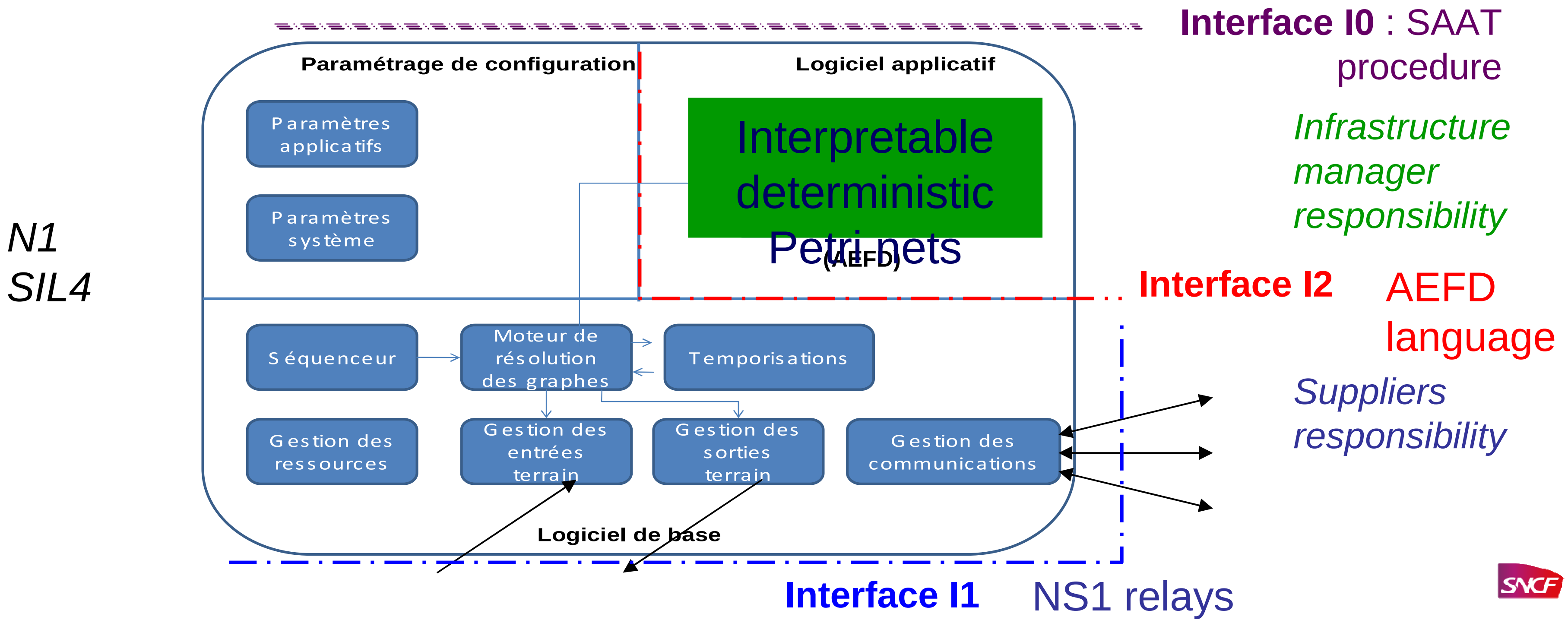
Railway characteristics

→ The SNCF designed PIPC interlocking system were designed:

- To carry out a clear separation between « hardware & basic software » (*suppliers view*) and « functional software » (*infrastructure manager view*)
- To carry out clear interfaces between the computerized module and rest of the railway system
- To carry out the specification and the functional software with interpretable deterministic Petri nets (*interpreted in the target machine*)
- To reduce the safety demonstration costs and to allow a formal validation of the functional software in the real environment conditions of the interlocking system
  ⇒ the method have to be applicable by signalling engineers

Railway characteristics

→ The architecture use common functional interfaces for all the interlocking systems (for all the suppliers)



Summary

Safety problems of IT-Systems

Railway characteristics

Interpretable deterministic Petri nets

Formal validation method

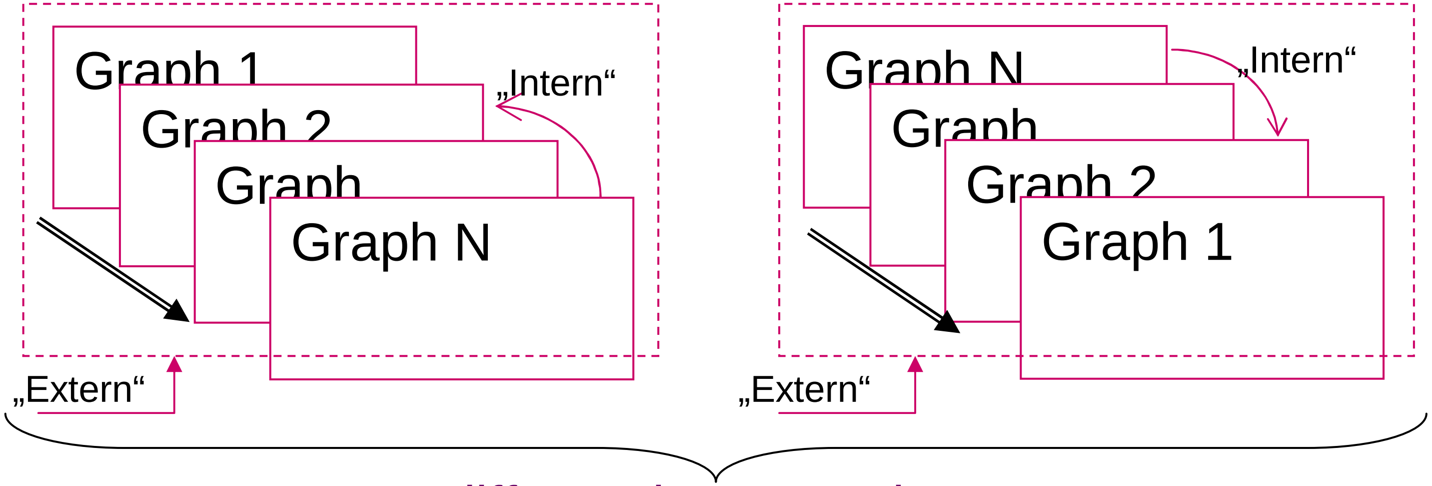
Application

Conclusion

Interpretable deterministic Petri Nets

→ The classical Petri nets aren't generally not interpretable in a deterministic way:

- It doesn't exist a distinction between „intern“ and „extern“ events
- It exist possible indecisions in the real time Petri nets interpretation (priorities…)

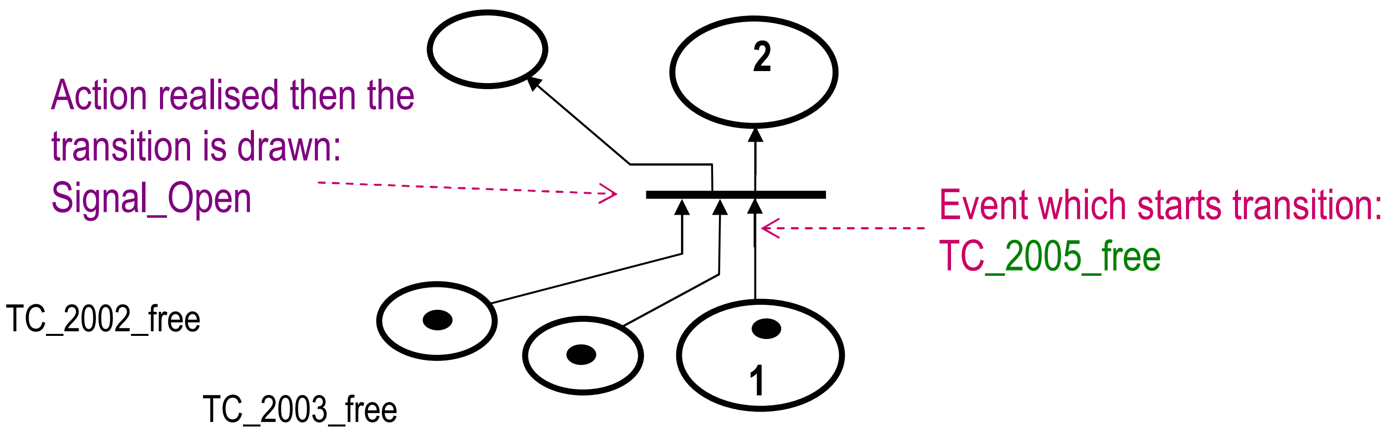


Two different interpretations
Two reachable system states trees

Interpretable deterministic Petri Nets

→ With classical Petri nets:
- The interpretation depends of the graph interpretation order
- The nets are generally not interpretable in real time

Classical PN

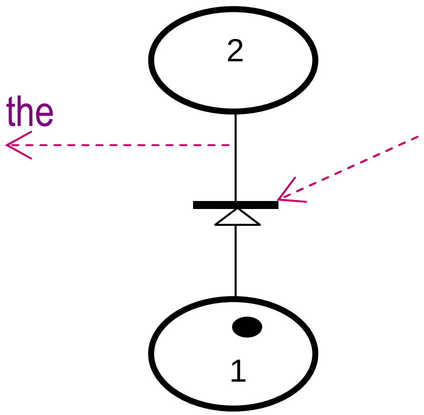


Interpretable deterministic Petri Nets

- AEFD language allows a deterministic functional specification and a deterministic interpretation of signalling functions (competing automats with constraints):
  - The interpretation is realisable without indecision
  - The interpretation is not dependant of the graphs reading order
  - The interpretation is realizable in real time

AEFD
Language

Action realised then the
transition is drawn:
Signal Open



Event which starts transition :
TC_2005_free
Condition : *TC_2002_free AND TC_2003_free*

Interpretable deterministic Petri Nets

→ AEFD definite language allows a deterministic functional specification and a deterministic interpretation of signalling functions:

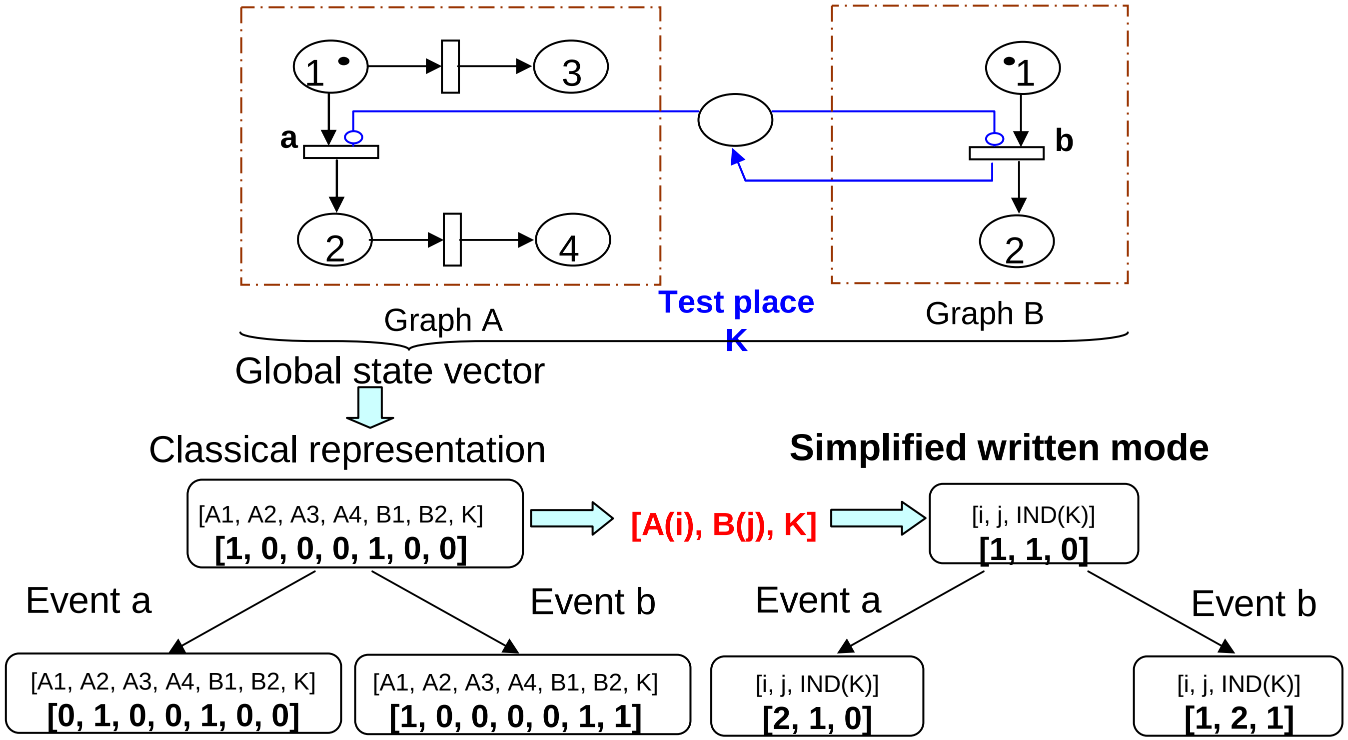
- The interpretation is realisable without indecision
- The interpretation is not dependant of the graphs reading order
- The interpretation is realizable in real time

Selected notation in the textual interpretable file form

```
…
Graph name
1
2
TC_2005_Libre Event
TC_2002_Libre AND TC_2003_Libre AND
TC 2005_Libre Condition
Signal_Open; Action
…
```

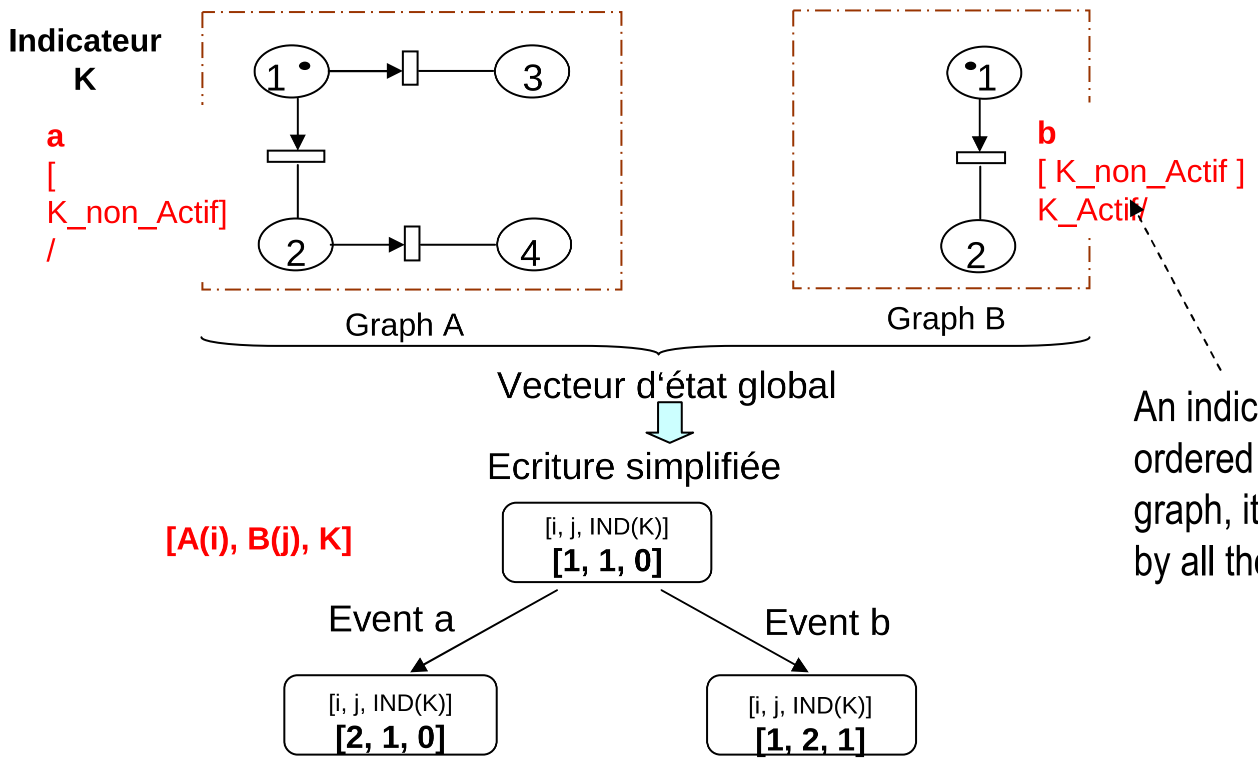
Interpretable deterministic Petri Nets

- Communication between graphs with classical Petri nets:



Interpretable deterministic Petri Nets

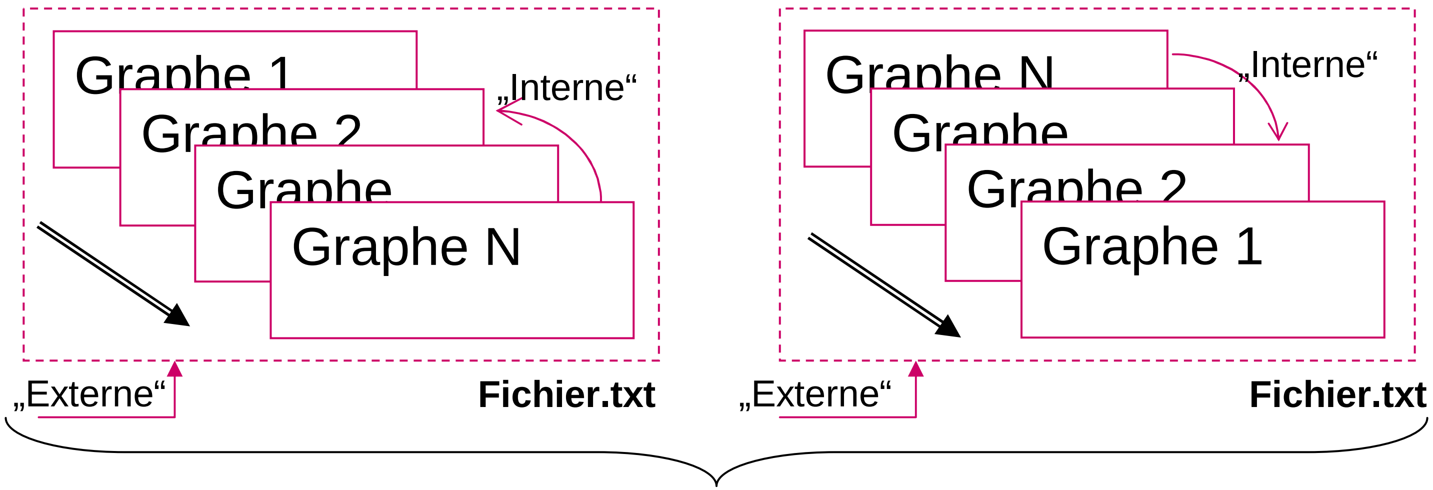
- Communication between graphs with the selected notation:



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

Interpretable deterministic Petri Nets

→ With the selected written mode, the Petri nets are interpretable in a deterministic way, without ambiguity and in real time



An unique reachable, finished and countable system states

Summary

Safety problems of IT-Systems

Railway characteristics

Interpretable deterministic Petri nets

Formal validation method

Application

Conclusion

Formal validation method

→ It exists two families of formal methods:

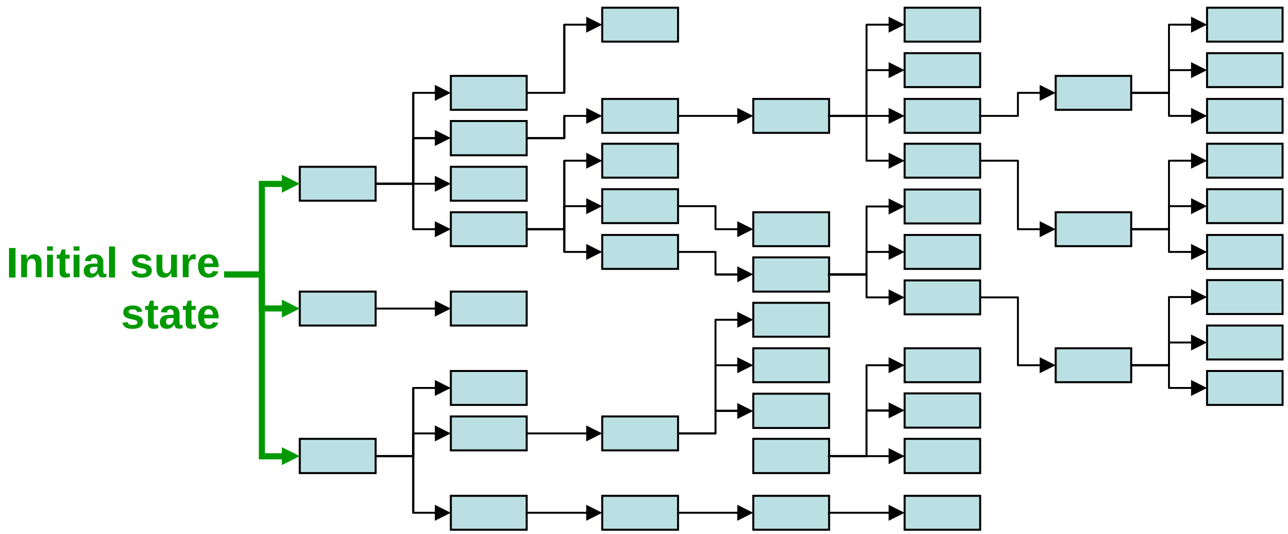
- → Formal design method:
  The proof is brought by code construction, the code is transcribed and compiled to be installed in the target machine (mainly a suppliers vision)
- → Formal validation method:
  The proof is brought on the final interpreted final functional model (mainly an infrastructure manager vision)

  The suggested method is a formal validation method

  The method is applicable on the functionalities written with deterministic and interpretable Petri nets

Formal validation method

→ The functions written with deterministic and interpretable PN can be represented by an unique reachable system states:



Systematically system states research

Post (Initial_state)*

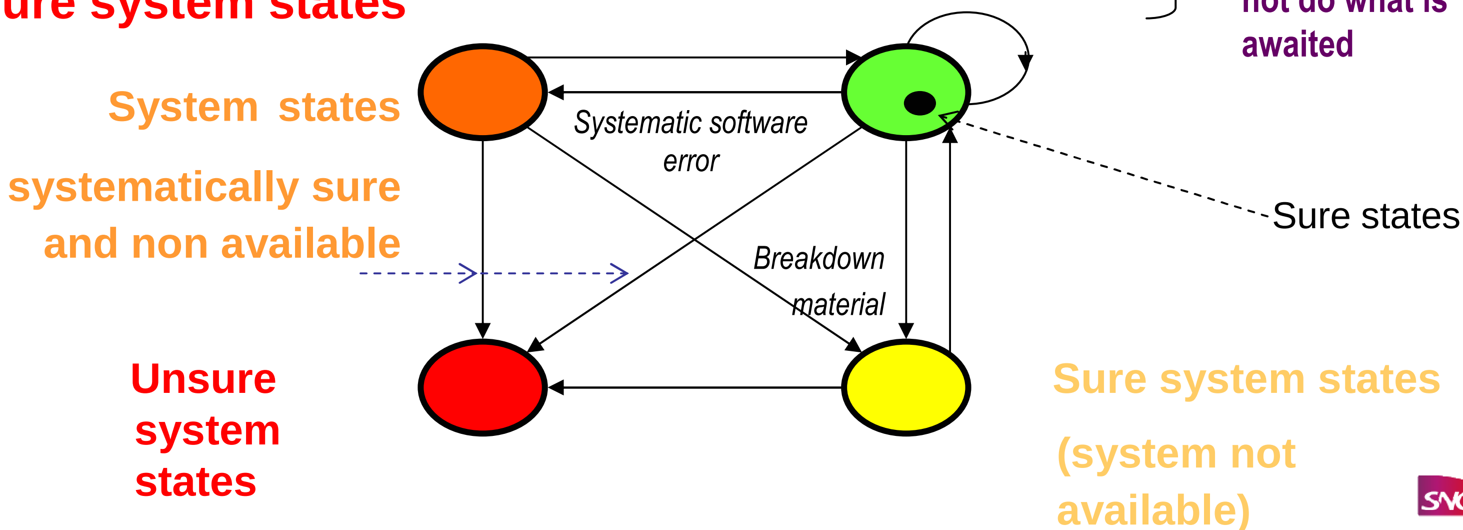
Formal validation method

→ Each state system can be associated with one with the 4 categories:

- System states sure and available
- System states sure but not available
- States system systematically reachable sure system states (not available)
- Unsure system states

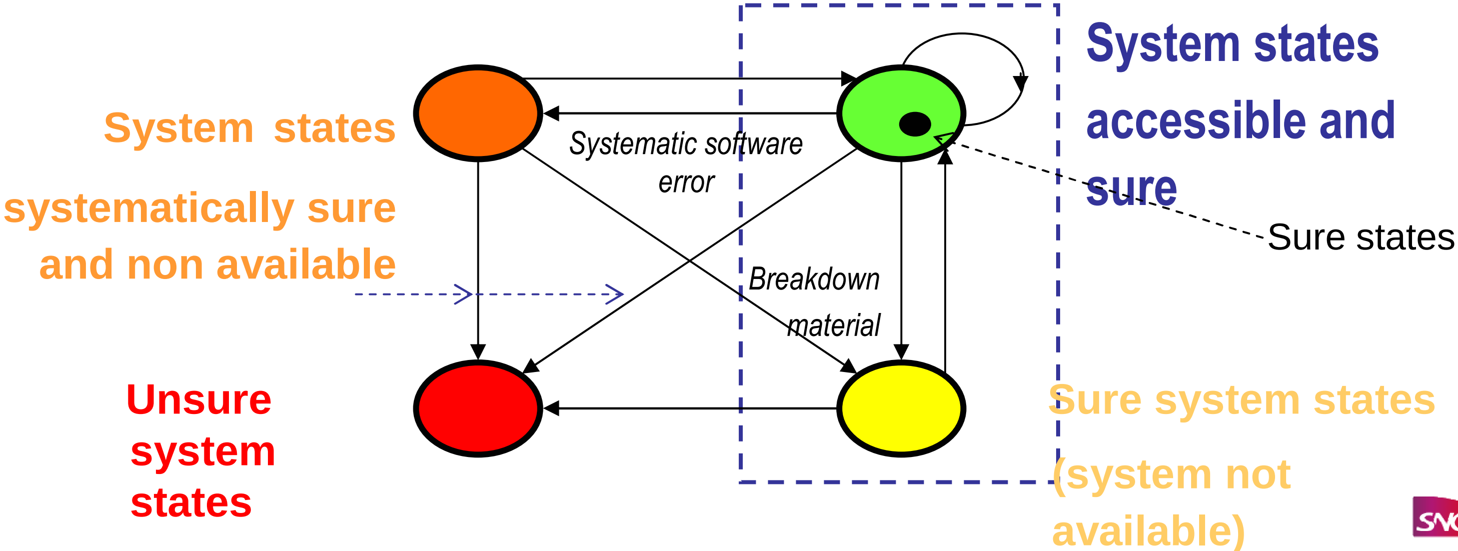
The system does what is awaited

The system doesn't not do what is awaited



Formal validation method

→ The safety properties must be written in order to be able to prove that no “sure but not available system state” (overabundant) or „unsure system state is reachable



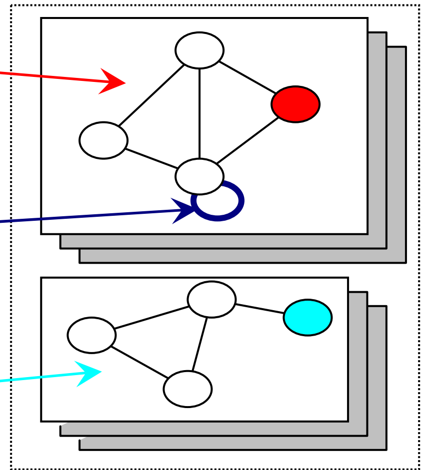
Formal validation method

→ The safety properties have to be written with « proof automats », by signalling engineers, in three stages:

Stage 1: description of the **safety properties or incompatibilities** they have to be ever respected by the railway system

Stage 2: description of the waited functionalities for the detection of « possible » **overabundant conditions**

Stage 3: **functional postulates** description (rules, environment…) limiting the validity field of the proof



Simple text file

Formal validation method

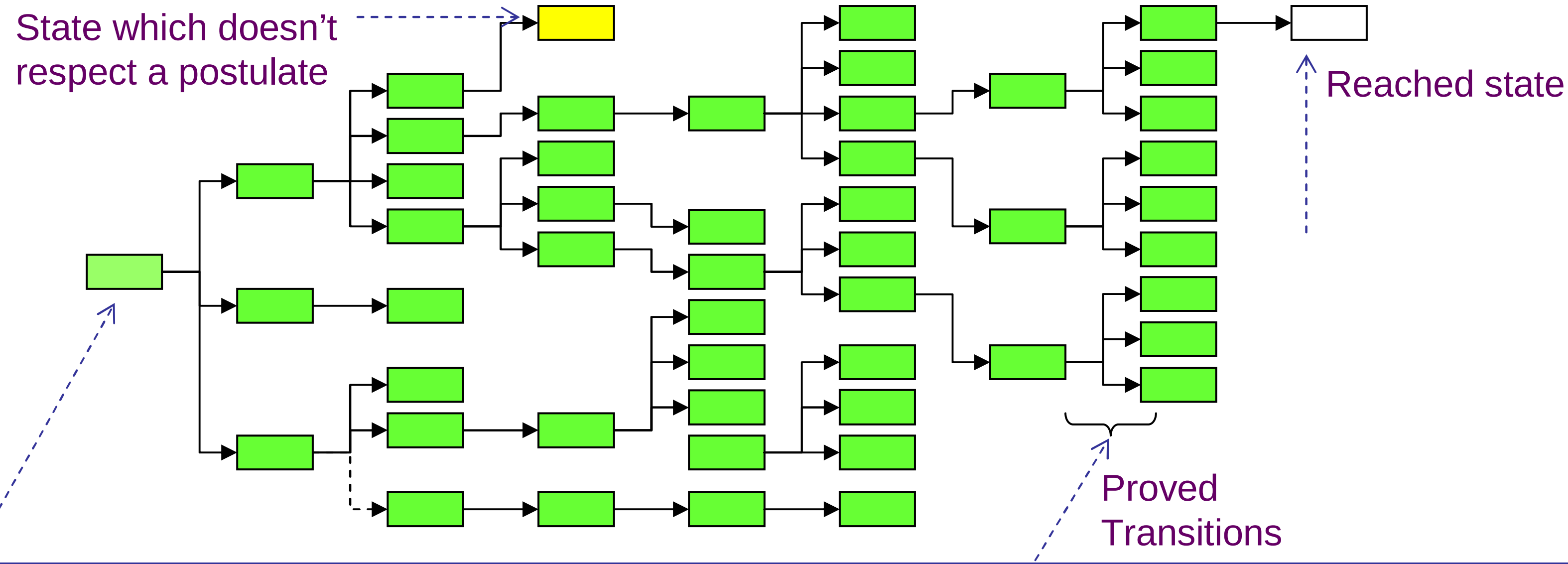
→ The proof can be accomplished in the following way with the use of the « functional graphs » and « proof graphs »:

$$\textbf{Post* (}\textbf{\textit{Etat Initial)}} \cap \textbf{Unsafe States} = \phi\ \textbf{?}$$

→ The proof principle is the following:
«If a group of properties is true for a given system state, and that this group remains proved during a transition between system states, then the property is true in the new system state»
This proof can be reproduced for every level of system states to the point of being applied by recurrence to all reachable system states. The initial state have to be safe.

Formal validation method

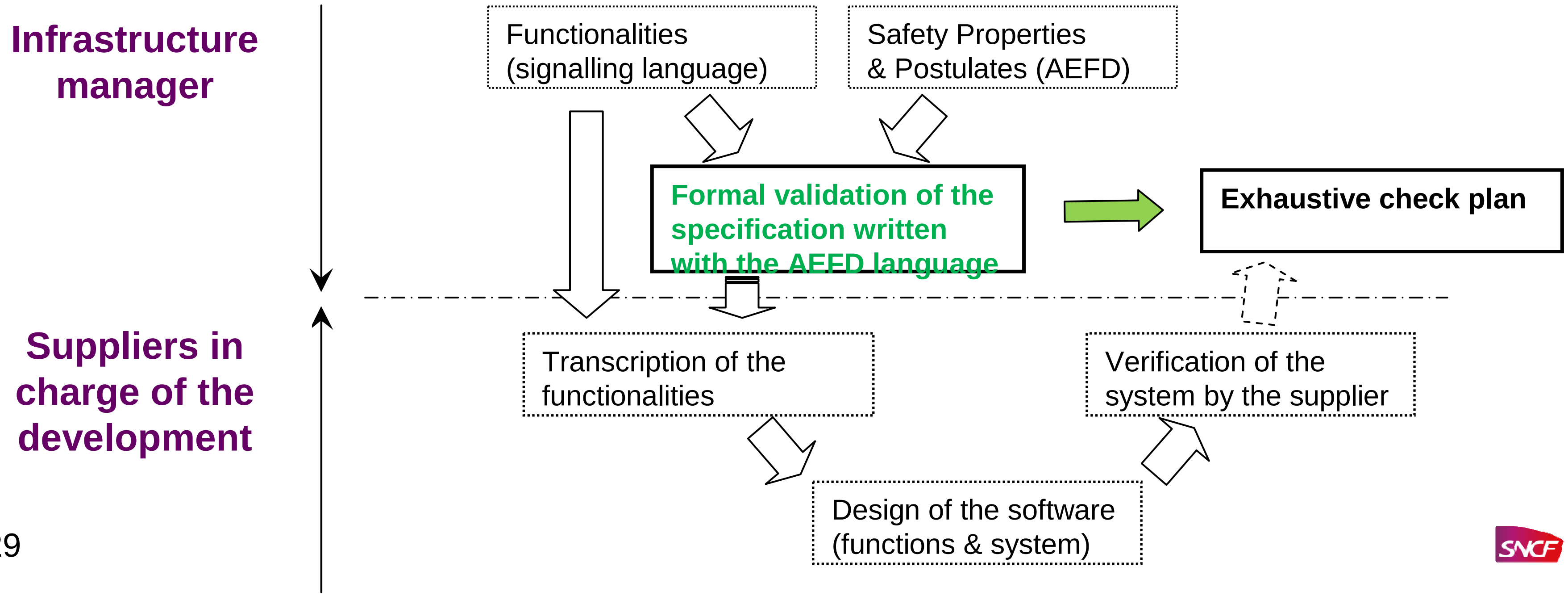
→ The basic principle is:



Initial safe state **AND** All the possible transitions are known **AND** All the reachable transitions are proved ⇒ All the reachable system states a safe

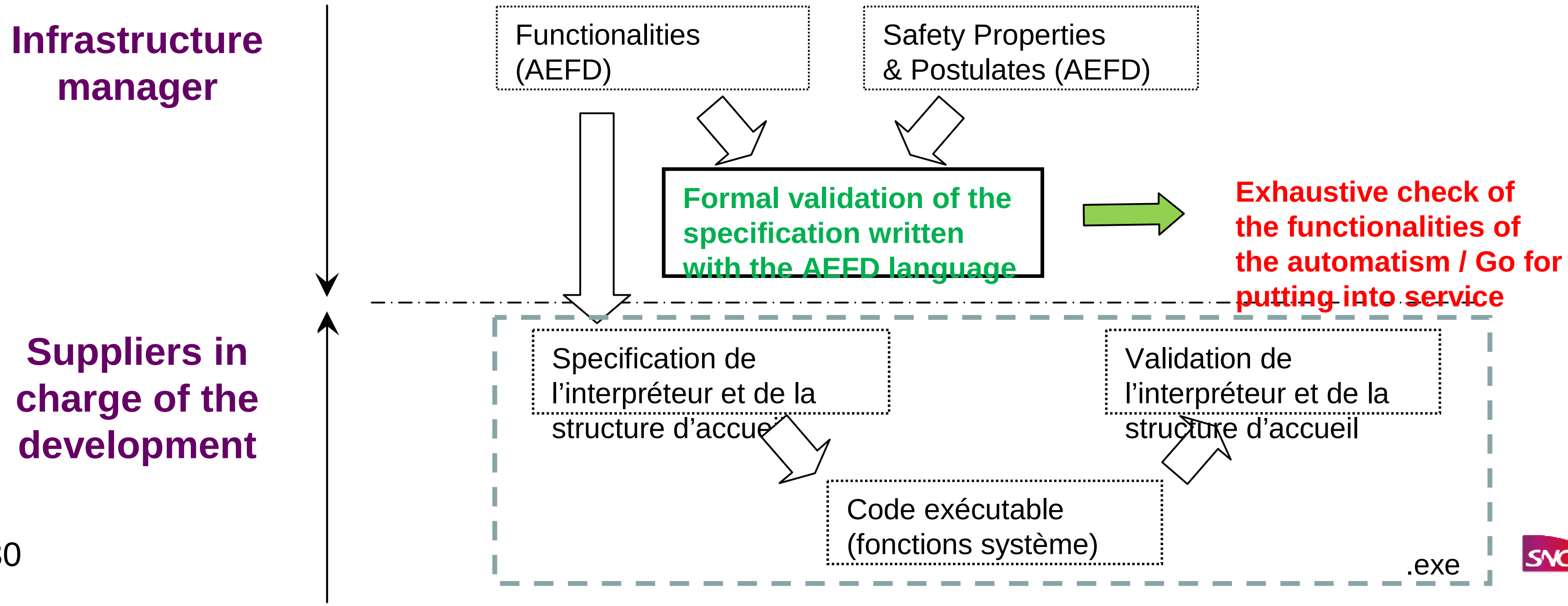
Formal validation method

- Use of the AEFD language as a unit specification language :
  1st use : proved specifications + exhaustive check plan generation



Formal validation method

- Use of the AEFD language as a unit specification language :
  2nd use : proved specifications + interpretation by a safe target unit



Summary

Safety problems of IT-Systems

Railway characteristics

Interpretable deterministic Petri nets

Formal validation method

Application

Conclusion

Application - Formal validation tools chain

Appropriated tools were developed by SNCF Infra to accomplish:

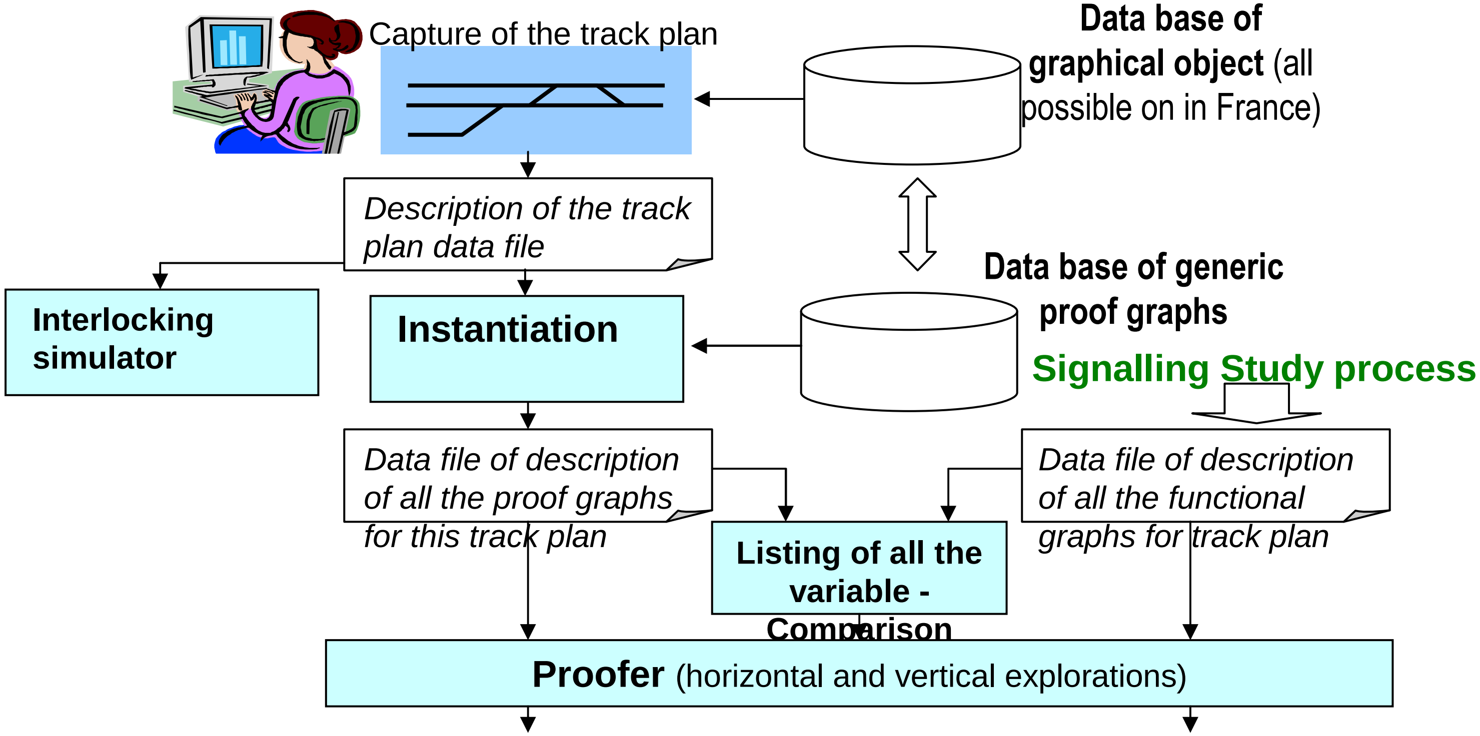
- Automatic definition of the safety properties and the postulates describing the conditions of use,
- Formal writing of these properties in order make the proof,
- Definition of the initial system state in which all the safety property are true,
- Evaluation of the safety properties by recurrence for 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** .

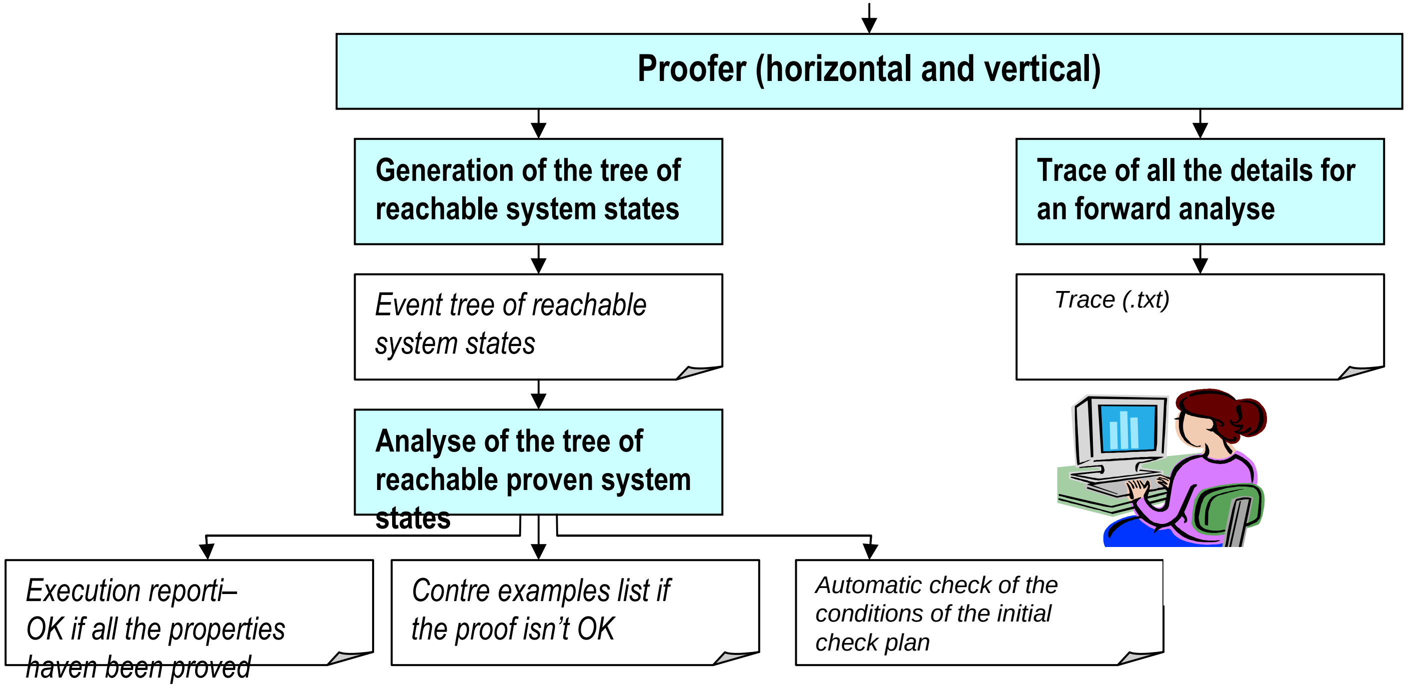
Application - Formal validation tools chain

- **Formal validation process - Step 1**



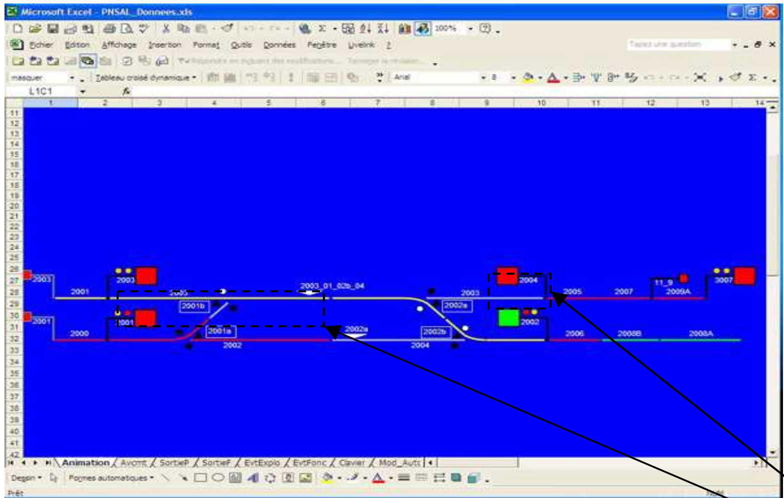
Application - Formal validation tools chain

- **Formal validation process - Step 2**



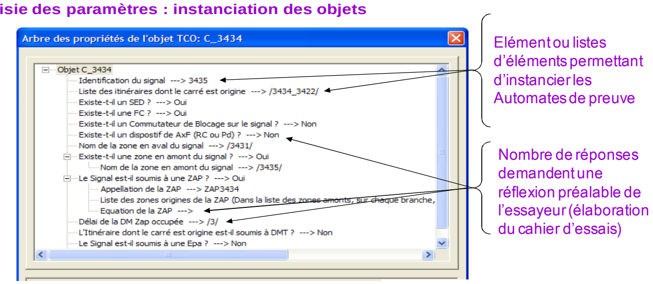
Application - Formal validation tools chain

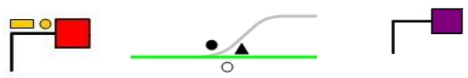
- **Track plan example and safety properties instantiation**



Capture of the track plan by topological association of graphical object

Saisie des paramètres : instanciation des objets

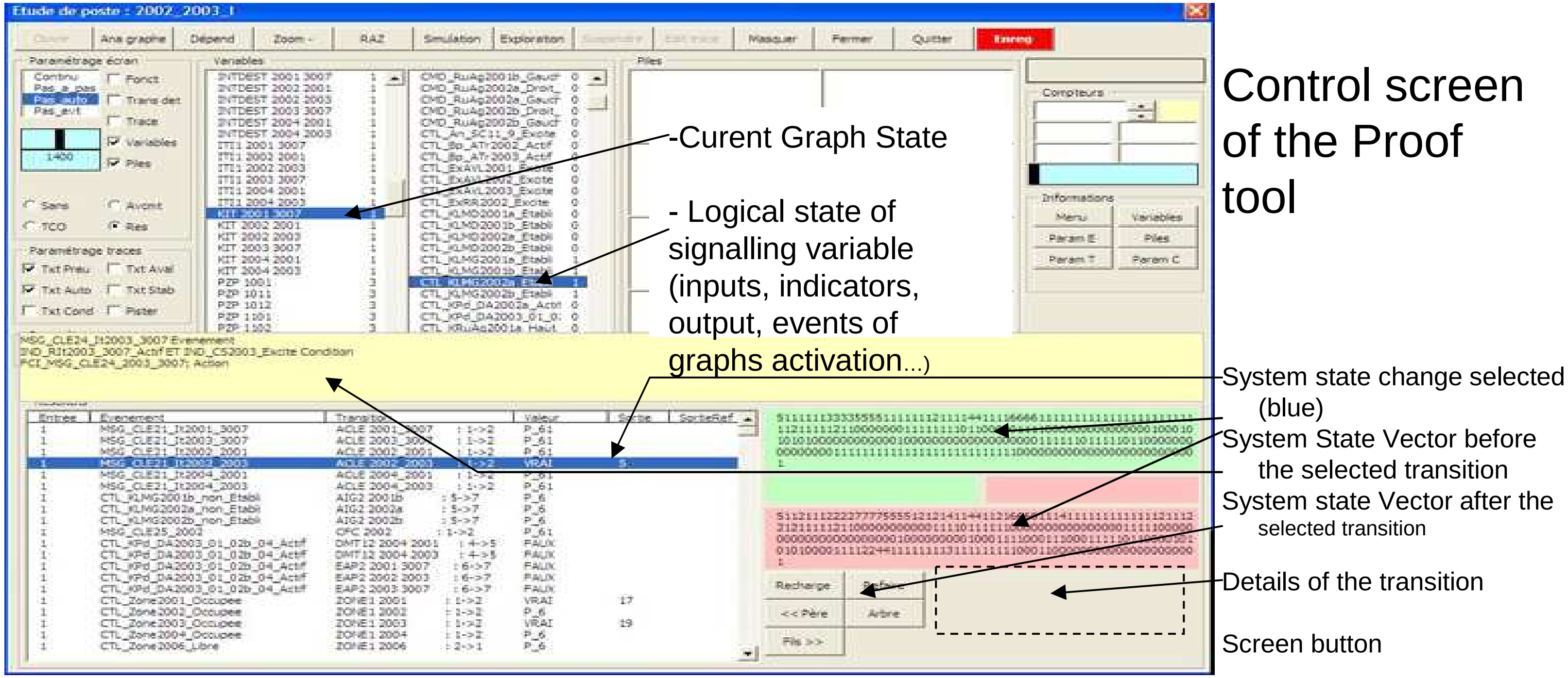




Graphical Objects topological laid out and instantiate: automatically or manually by the signalling engineer in charge of the proof:
- Signal object,
- Switch object…

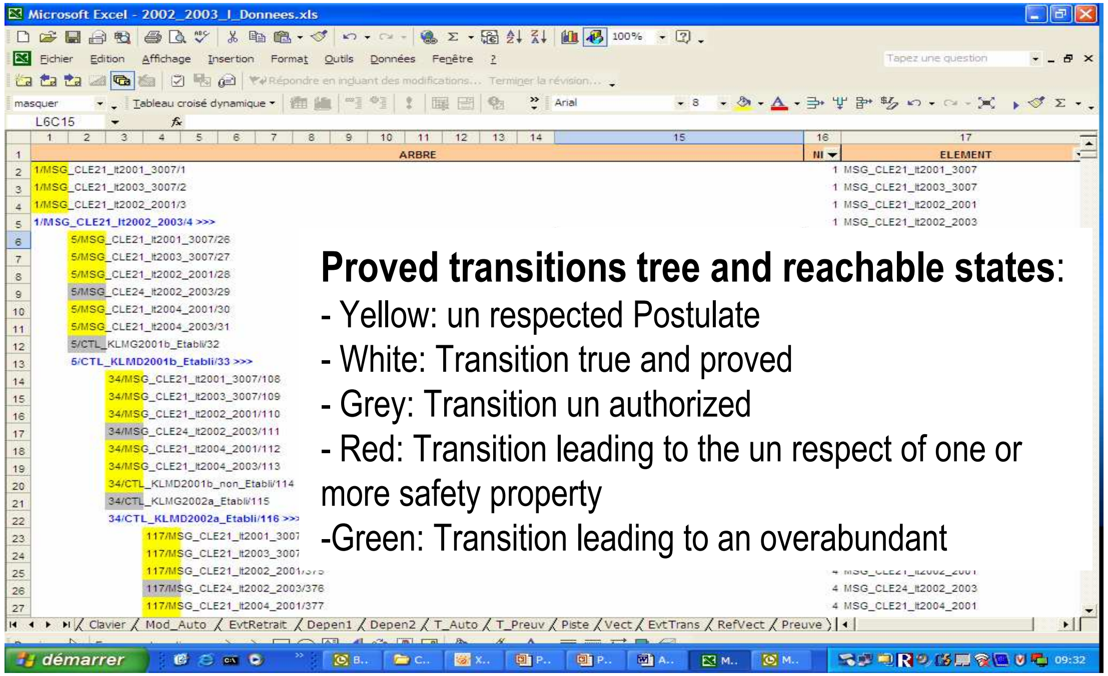
Application - Formal validation tools chain

- **Proof tool view**



Application - Formal validation tools chain

- **Reachable states tree tool view**



Proved transitions tree and reachable states:

- Yellow: un respected Postulate
- White: Transition true and proved
- Grey: Transition un authorized
- Red: Transition leading to the un respect of one or more safety property
-Green: Transition leading to an overabundant

(1) To carry out the vivacity check

(2) To carry out the execution report

(3) To presenter the results with ergonomic manner

(4) To carry out the tree of the transitions tree

Summary

Safety problems of IT-Systems

Railway characteristics

Interpretable deterministic Petri nets

Formal validation method

Application

Conclusion

Conclusion

→ The development of critical computerized systems should not take place any more without application of a formal method allowing to guarantee the functional software:

- In particular for the system “*to complicated to be tested*”…

→ The practical application of formal methods requires to create from the design the necessary conditions for its realization:

- **The safety properties can't be written by suppliers or mathematicians**, but only by Signalling men : the only persons who know the postulates of the system, the environment conditions…
- **It is necessary to differentiate clearly the <u>functional software</u>** (signalling) **and the <u>basic software</u>** (computer science)

Conclusion

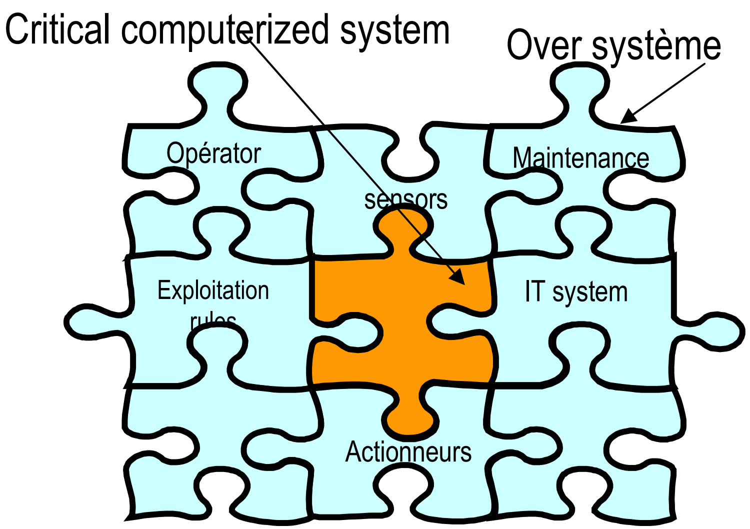
→ **The method is applied with functional software** defined with deterministic and interpretable Petri nets. It key points are:

- Model based specifications, provable and interpretable in real time, can be used for critical IT-Systems (300 in use today)
- No risk of error introduction during the code generation and compilation
- Less expensive than tests accomplished traditionally
- The infrastructure manager controls the functionalities… with his own people
- Can be used in an industrial way, without people educated in mathematics,
- Automatic and exhaustive check of the interlocking system
- Is now applied on a real interlocking systems

→ ***The real difficulty is the generic identification and the formalization of safety properties and postulates***

Conclusion

The method allows to realize industrially a formal <u>validation</u> 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

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. It will be used by the SNCF Infra and the UIC

The application of formal methods is now an obligation for the development of new critical IT system if we 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.

Thank you for your attention
Any question?

→ Formal proven since 1896

Dr Marc ANTONI
SNCF – Infrastructure Direction
marc.antoni@sncf.fr

Because you will never have the possibility to come back and try again…

Dr Marc ANTONI
SNCF – Infrastructure Direction
marc.antoni@sncf.fr