

# Formal Methods Outside the Mother Land

Aryldo G Russo Jr.

AeS Group & Research Institute of State of São Paulo (IPT)

July 17, 2009



# Agenda

- 1 Introduction
- 2 Where formal methods (could be) are used
- 3 Tool comparison
- 4 Experiences
- 5 Gaps or needs
- 6 Conclusion



# Introduction

- The primary objective of this paper is to present the current State of Practice of Formal Methods in countries outside Europe, namely, Brazil and Korea.



# Introduction

- The primary objective of this paper is to present the current State of Practice of Formal Methods in countries outside Europe, namely, Brazil and Korea.
- Initially, a background information about the reason to start working with Formal Methods, and the involvement of AeS group with academia is given.



# Introduction

- The primary objective of this paper is to present the current State of Practice of Formal Methods in countries outside Europe, namely, Brazil and Korea.
- Initially, a background information about the reason to start working with Formal Methods, and the involvement of AeS group with academia is given.
- Then, a general scenario of how these methods are being used nowadays in Brazil and Korea and particularly some industrial areas where formal methods are currently applied are shown.

# Introduction

- Finally, a comparison of three tools, namely, AtelierB[1], RODIN[2] and SCADE[3] is presented.



# Introduction

- Finally, a comparison of three tools, namely, AtelierB[1], RODIN[2] and SCADE[3] is presented.
- At the end, the author presents some gaps that, from his personal point of view, can be fulfilled with some new or in phase of development, plugins and language extensions.



# Aes Group

- The AeS Group has developed railway sub-systems since 1998.





# Aes Group

- The AeS Group has developed railway sub-systems since 1998.
- Door system became one of the most important in the railway market



# Aes Group

- The AeS Group has developed railway sub-systems since 1998.
- Door system became one of the most important in the railway market
- AeS Group has acquired a reputation as a company that has the needed know-how to develop safety critical applications.

# Aes Group

- The AeS Group has developed railway sub-systems since 1998.
- Door system became one of the most important in the railway market
- AeS Group has acquired a reputation as a company that has the needed know-how to develop safety critical applications.
- AeS group decided to identify a formal method that would best fit the current CGP SIL 3-level requirements and railway industry standard practices and standards (as is the case of CENELEC EN 50128[4]).



# Aes Group

- AeS group decided, first, to study and use the B method[5] and, second, to look for assistance from academia, which was obtained from two Brazilian Universities (Universidade de São Paulo and Universidade do Rio Grande do Norte).

# Aes Group

- AeS group decided, first, to study and use the B method[5] and, second, to look for assistance from academia, which was obtained from two Brazilian Universities (Universidade de São Paulo and Universidade do Rio Grande do Norte).
- Nowadays, AeS Group has also the support of DEPLOY project and some universities like University of Southampton, and University of York, besides companies like ClearSy and Esterel.



# Technological Research Institute of the State of São Paulo (IPT)

- The author decided to finally initiate a "formal" dedication in the Formal Methods field, and have chosen the Technological Research Institute of State of São Paulo (IPT) as starting point.



# Technological Research Institute of the State of São Paulo (IPT)

- The author decided to finally initiate a "formal" dedication in the Formal Methods field, and have chosen the Technological Research Institute of State of São Paulo (IPT) as starting point.
- In the mean time, the author joined the Software Requirements Specification Laboratory (SoftREL).



# General scenario

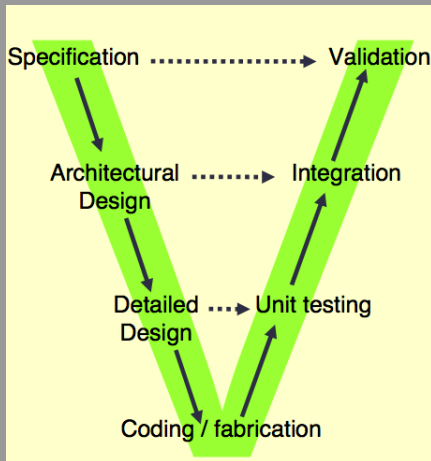


Figure: V Model - Software Development Model



# General scenario

- The software development process presented in the IEC 61508[6] is well known in South American companies, but those recommendations are frequently put aside.

# General scenario

- The software development process presented in the IEC 61508[6] is well known in South American companies, but those recommendations are frequently put aside.
- This is a good scenario to try to better the process through the use formal methods without changing the manual tasks.

# General scenario

- The software development process presented in the IEC 61508[6] is well known in South American companies, but those recommendations are frequently put aside.
- This is a good scenario to try to better the process through the use formal methods without changing the manual tasks.
- Those processes are barely known at the Far East.

# Companies in South America

- They are aware of the whole process

# Companies in South America

- They are aware of the whole process
- They usually rely on tests to guarantee the expected behavior



# Companies in South America

- They are aware of the whole process
- They usually rely on tests to guarantee the expected behavior
- The transition takes place directly from NL specification to the code phases, some times, through an intermediate phase, based usually in UML specifications.

# Companies in South America

- They are aware of the whole process
- They usually rely on tests to guarantee the expected behavior
- The transition takes place directly from NL specification to the code phases, some times, through an intermediate phase, based usually in UML specifications.
- apparently, there is no traceability methodology

## Companies in the Far East - South Korea

- The coding phase starts right after receiving the specification





## Companies in the Far East - South Korea

- The coding phase starts right after receiving the specification
- Quality is the main concern, but no defined process is used to ensure that



## Companies in the Far East - South Korea

- The coding phase starts right after receiving the specification
- Quality is the main concern, but no defined process is used to ensure that
- They rely on experienced professionals to reach the desired quality level

## Companies in the Far East - South Korea

- The coding phase starts right after receiving the specification
- Quality is the main concern, but no defined process is used to ensure that
- They rely on experienced professionals to reach the desired quality level
- clearly, there are only three phases: specification, coding and integration tests



# General scenario

- Based on this view, it becomes clear that formal methods can not be applied straightforward.



# General scenario

- Based on this view, it becomes clear that formal methods can not be applied straightforward.
- Before, it is necessary to create a better culture on software development process.



## Where formal methods (could be) are used

- Many different industrial areas, such as, Nuclear[7], Medical devices[8], Avionics, Aerospace and transportation [9].



## Where formal methods (could be) are used

- Many different industrial areas, such as, Nuclear[7], Medical devices[8], Avionics, Aerospace and transportation [9].
- Some examples are the emergency contention measures in nuclear power plants, health support devices in medical applications, automatic pilot on avionics, positioning systems in aerospace and signaling systems in transportation.

# Where formal methods (could be) are used

- Many different industrial areas, such as, Nuclear[7], Medical devices[8], Avionics, Aerospace and transportation [9].
- Some examples are the emergency contention measures in nuclear power plants, health support devices in medical applications, automatic pilot on avionics, positioning systems in aerospace and signaling systems in transportation.
- There is plenty of space for the adoption of supporting tools.



## Where formal methods (could be) are used

- The application of formal methods in these fields is not the true reality in South America and Far East.



## Where formal methods (could be) are used

- The application of formal methods in these fields is not the true reality in South America and Far East.
- The distance between mathematical notation and the normal procedures used so far has to be shortened.



## Where formal methods (could be) are used

- The application of formal methods in these fields is not the true reality in South America and Far East.
- The distance between mathematical notation and the normal procedures used so far has to be shortened.
- The B formal method is the most frequently used in railway field.



## Where formal methods (could be) are used

- The application of formal methods in these fields is not the true reality in South America and Far East.
- The distance between mathematical notation and the normal procedures used so far has to be shortened.
- The B formal method is the most frequently used in railway field.
- Recently, the Esterel formal method began to be used as well.

## Where formal methods (could be) are used

- The application of formal methods in these fields is not the true reality in South America and Far East.
- The distance between mathematical notation and the normal procedures used so far has to be shortened.
- The B formal method is the most frequently used in railway field.
- Recently, the Esterel formal method began to be used as well.
- In any field of application, formal methods, and their related tools, can help in the development process replacing the human interaction.



# Tool comparison

- A brief comparison of some existent tools is presented.

# Tool comparison

- A brief comparison of some existent tools is presented.
- Those tools are, Atelier B, RODIN and SCADE.

# Methodology

- The tools are classified according MY personal feelings based on comments obtained in trainings during the last 3 years.



# Methodology

- The tools are classified according MY personal feelings based on comments obtained in trainings during the last 3 years.
- The comparison methodology was based on three aspects, as follows:
  - *capability*: the verification of how these tools can satisfy project constraints
  - *usability*: basically, which is the difficulty the user faces when trying to use the tool
  - *adequacy to the current development process* : how the tool can better fit in the process without causing too many changes in the way it was performed so far



# Methodology

- To make a classification of these aspects I used a simple ranking method, as follows:
  - 1 Very difficult
  - 2 Medium
  - 3 easy

# results

Aspect	<b>capability</b>	<b>usability</b>	<b>adaptation</b>	<b>Results</b>
AtelierB	2	1	2	5
RODIN	2	2	1	5
SCADE	2	3	3	8

Table: Comparison table

# Justification

- AtelierB
  - the capability to solve the project constraints is not so bad, but it is necessary to know a lot of the formal language and constructs to be able to have easy proof obligations.

# Justification

- AtelierB
  - the capability to solve the project constraints is not so bad, but it is necessary to know a lot of the formal language and constructs to be able to have easy proof obligations.
  - although, the version 4 of AtelierB supplies a real better usability, all comments received so far are based on the previous version where the lack of a good User Interface makes its usage painful.

# Justification

- AtelierB
  - the capability to solve the project constraints is not so bad, but it is necessary to know a lot of the formal language and constructs to be able to have easy proof obligations.
  - although, the version 4 of AtelierB supplies a real better usability, all comments received so far are based on the previous version where the lack of a good User Interface makes its usage painful.
  - since it allows to go from the specification to the code it can be considered as a good tool for that purpose, but as the interactions during the middle phases (refinements) are some times, painful, it can not receive the higher grade.

# Justification

- RODIN
  - since it is not so different from AtelierB, similar results are shown, i.e. the capability to solve the project constraints is not so bad, but it is necessary to know a lot of the formal language and constructs to be able to have easy proof obligations.

# Justification

- RODIN
  - since it is not so different from AtelierB, similar results are shown, i.e. the capability to solve the project constraints is not so bad, but it is necessary to know a lot of the formal language and constructs to be able to have easy proof obligations.
  - the way that RODIN was constructed is quite helpful for a non experienced person, as it is only necessary to fill down some fields to have a basic specification, but the lack of text editor that could help more experienced person and speed up the specification process lowers its classification



# Justification

## ■ RODIN

- since it is not so different from AtelierB, similar results are shown, i.e. the capability to solve the project constraints is not so bad, but it is necessary to know a lot of the formal language and constructs to be able to have easy proof obligations.
- the way that RODIN was constructed is quite helpful for a non experienced person, as it is only necessary to fill down some fields to have a basic specification, but the lack of text editor that could help more experienced person and speed up the specification process lowers its classification
- the lack of possibilities of decomposition at the moment of the evaluation and the ability to help only in the system specification phase, make of RODIN a difficult tool to be used in the current process.

# Justification

- SCADE
  - even based on a different concept, where formal methods are behind the scene, it has a great capability to deal with project constraints, but some formal background is still needed to construct correct models.

# Justification

- SCADE
  - even based on a different concept, where formal methods are behind the scene, it has a great capability to deal with project constraints, but some formal background is still needed to construct correct models.
  - as it was built from the very beginning to be an industrial tool; its usability is its strongest point, with a good interface and a lot of fancy features that captivate the user. A lot of things can be done based on templates and patters, what helps a lot as well

# Justification

## ■ SCADE

- even based on a different concept, where formal methods are behind the scene, it has a great capability to deal with project constraints, but some formal background is still needed to construct correct models.
- as it was built from the very beginning to be an industrial tool; its usability is its strongest point, with a good interface and a lot of fancy features that captivate the user. A lot of things can be done based on templates and patters, what helps a lot as well
- Besides the capability to go from the specification to the code, it has also some other complementary tools which help in important auxiliary tasks in the project such as requirement management, traceability, etc..

# Experiences

- Experiences in introduction of formal methods in the last 3 years.

# Experiences

- Experiences in introduction of formal methods in the last 3 years.
- A basic conclusion was that even if formal methods can not fulfill all industrial needs they can help a lot to better model the development process and the resultant product (or software).

# Experiences

- Experiences in introduction of formal methods in the last 3 years.
- A basic conclusion was that even if formal methods can not fulfill all industrial needs they can help a lot to better model the development process and the resultant product (or software).
- 3 different examples based on different approaches.



# Signaling system

- Railway European companies are known as some of a few that use formal methods during the development process.



# Signaling system

- Railway European companies are known as some of a few that use formal methods during the development process.
- It is also true that, not all of their branches around the world follow the same concept.



# Signaling system

- Railway European companies are known as some of a few that use formal methods during the development process.
- It is also true that, not all of their branches around the world follow the same concept.
- During 2008, I participated in a revalidation process of a signaling system using B method and its associate tool, AtelierB.



# Signaling system

- The new project was based on a previous one.



# Signaling system

- The new project was based on a previous one.
- The task consisted in implementing new functions and then revalidating all the system



# Signaling system

- The new project was based on a previous one.
- The task consisted in implementing new functions and then revalidating all the system
- The changes were applied in the abstract model, and after that they were reflected in the refinements and implementation.

# Signaling system

- The new project was based on a previous one.
- The task consisted in implementing new functions and then revalidating all the system
- The changes were applied in the abstract model, and after that they were reflected in the refinements and implementation.
- New proof obligations were generated and the affected older ones were reapplied.



## Signaling system - results

- No failures where detected after the deployment of the system.

# Signaling system - results

- No failures were detected after the deployment of the system.
- The associated costs in this development were less than in a traditional process as:
  - there were no needs of maintenance changes
  - the necessary time dedicated to testing was really short.



# Signaling system - results

- No failures were detected after the deployment of the system.
- The associated costs in this development were less than in a traditional process as:
  - there were no needs of maintenance changes
  - the necessary time dedicated to testing was really short.
- But this job was performed for a company that has been using formal methods for a long time.

# Door system

- Verify the consistency of a door system specification.

# Door system

- Verify the consistency of a door system specification.
- RODIN was used as a proof of concept.

# Door system

- Verify the consistency of a door system specification.
- RODIN was used as a proof of concept.
- The objective was to help the door system manufacturer to rewrite the specification based on the result of the verification of the formal model.

# Door system

- Verify the consistency of a door system specification.
- RODIN was used as a proof of concept.
- The objective was to help the door system manufacturer to rewrite the specification based on the result of the verification of the formal model.
- The natural language specification is more than 100 pages long, and the needed information is spread out over all this specification.

# Door system

- The following two statements of the specification show one of the contradictions that were found
  - The train is not allowed to move while at least one door is open;
  - If the emergency button is pressed, the respective door must open when the train speed is under 6 km/h.

# Door system

- The following two statements of the specification show one of the contradictions that were found
  - The train is not allowed to move while at least one door is open;
  - If the emergency button is pressed, the respective door must open when the train speed is under 6 km/h.
- In this example it is easy to notice the contradiction, but those statements were spread out in the specification, so the direct comparison was not so clear.

# Door system

- The following two statements of the specification show one of the contradictions that were found
  - The train is not allowed to move while at least one door is open;
  - If the emergency button is pressed, the respective door must open when the train speed is under 6 km/h.
- In this example it is easy to notice the contradiction, but those statements were spread out in the specification, so the direct comparison was not so clear.
- In the example, the contradiction refers to the behavior of the door, which should not open until the train is completely stopped, but which also should open in an emergency situation when the speed of the train is under 6 km/h.



# Door system - Event B model

**MACHINE** Open\_contradiction

## VARIABLES

`train_stoped` boolean. when the train is stoped  
it's value is TRUE

`train_low_speed` boolean. when the train speed  
is below 6km/h it value is TRUE

`door_authorization` boolean. when the train is  
allowed to open doors it's value is  
TRUE

`emergency_buttom` boolean. if the buttom is  
pressed, it's value is TRUE

`open_comand` boolean. if true, command the  
opening

`train_speed` NAT. real speed

# Door system - Event B model

## INVARIANTS

*inv1* :  $train\_stoped \in \text{BOOL}$

*inv2* :  $door\_authorization \in \text{BOOL}$

*inv3* :  $train\_low\_speed \in \text{BOOL}$

*inv4* :  $emergency\_buttom \in \text{BOOL}$

*inv5* :  $train\_stoped =$

$\text{TRUE} \Rightarrow door\_authorization = \text{TRUE}$

*inv6* :  $train\_stoped =$

$\text{FALSE} \Rightarrow door\_authorization = \text{FALSE}$

*inv7* :  $train\_stoped =$

$\text{TRUE} \Rightarrow train\_low\_speed = \text{TRUE}$

*inv9* :  $open\_comand \in \text{BOOL}$

*inv10* :  $train\_speed \in \mathbb{N}$

*inv11* :  $door\_authorization =$

$\text{FALSE} \Rightarrow open\_comand = \text{FALSE}$



# Door system - Event B model

## EVENTS

### Initialisation

begin

```
act1 : door_authorization :=  
      TRUE  
act2 : train_stoped := TRUE  
act3 : train_low_speed := TRUE  
act4 : emergency_buttom :=  
      FALSE  
act5 : open_comand := FALSE  
act6 : train_speed := 0
```

end



# Door system - Event B model

```
Event EMERGENCY_OPEN ≐  
  when  
    grd1 : train_low_speed = TRUE  
    grd3 : emergency_button =  
           TRUE  
  then  
    act1 : open_comand := TRUE  
  end
```

# Door system - Event B model

```
Event LOW_SPEED_MONITOR  $\hat{=}$   
    when  
        grd1 : train_speed  $\leq$  6  
    then  
        act1 : train_low_speed := TRUE  
    end
```

# Door system - Event B model

```
Event ZERO_SPEED_MONITOR  $\hat{=}$   
  when  
    grd1 : train_speed = 0  
  then  
    act1 : train_stoped := TRUE  
    act2 : train_low_speed := TRUE  
    act3 : door_authorization :=  
           TRUE  
  end
```

# Door system - Event B model

```
Event AUTHORIZARION_RELEASE  $\hat{=}$   
  when  
    grd1 : train_speed > 0  
  then  
    act1 : door_authorization :=  
            FALSE  
    act2 : train_stoped := FALSE  
    act3 : open_comand := FALSE  
  end
```

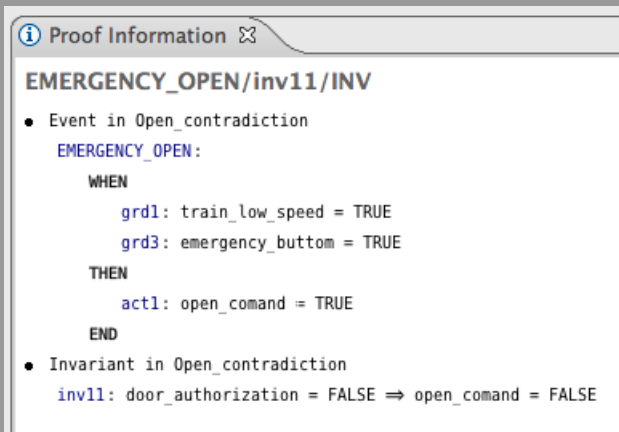
# Door system - Event B model

```
Event LOW_SPEED_RELEASE  $\hat{=}$   
    when  
        grd1 : train_speed > 6  
    then  
        act1 : train_low_speed :=  
                FALSE  
        act2 : train_stoped := FALSE  
        act3 : door_authorization :=  
                FALSE  
        act4 : open_comand := FALSE  
    end  
END
```



## Door system - Results

It's clear that to discharge this PO, (figure 2) it is not a question of correcting the model, but the natural language specification must be changed to avoid this kind of ambiguities or contradictions.



**Proof Information** ✖

**EMERGENCY\_OPEN/inv11/INV**

- Event in Open\_contradiction  
**EMERGENCY\_OPEN:**  
    **WHEN**  
        **grd1:** train\_low\_speed = TRUE  
        **grd3:** emergency\_button = TRUE  
    **THEN**  
        **act1:** open\_comand = TRUE  
    **END**
- Invariant in Open\_contradiction  
    **inv11:** door\_authorization = FALSE  $\Rightarrow$  open\_comand = FALSE

# Door system - Results

In this case three different approaches or options were proposed, as follows:

- 1 The train is not allowed to move when at least one door is open, *unless in a emergency situation*;

The first option was chosen by the customer, and the specification and model were changed to reflect this new constraint.



# Door system - Results

In this case three different approaches or options were proposed, as follows:

- 1 The train is not allowed to move when at least one door is open, *unless in a emergency situation*;
- 2 The train is not allowed to move *over 6 km/h* when at least one door is open;

The first option was chosen by the customer, and the specification and model were changed to reflect this new constraint.



## Door system - Results

In this case three different approaches or options were proposed, as follows:

- 1 The train is not allowed to move when at least one door is open, *unless in a emergency situation*;
- 2 The train is not allowed to move *over 6 km/h* when at least one door is open;
- 3 If the emergency button is pressed, the respective door must open when the train *stops*

The first option was chosen by the customer, and the specification and model were changed to reflect this new constraint.

## Door system - Results

- This simple example helped to present the formal method benefits, stating the impossibility to introduce ambiguities and contradictions.

# Door system - Results

- This simple example helped to present the formal method benefits, stating the impossibility to introduce ambiguities and contradictions.
- The objective now is:
  - Try to represent the complete specification of one train sub-system
  - Reformulate the natural language specification in a better representation.
  - Pointing out the items that need to be revised to create a more consistent specification.

# Platform screen doors

- Platform Screen Doors, aka PSD, is a door system that is installed in the platform stations to avoid people to fall down to the track.



# Platform screen doors

- Platform Screen Doors, aka PSD, is a door system that is installed in the platform stations to avoid people to fall down to the track.
- The safety related issues are even higher than for the train door system.





# Platform screen doors

- Platform Screen Doors, aka PSD, is a door system that is installed in the platform stations to avoid people to fall down to the track.
- The safety related issues are even higher than for the train door system.
- This kind of system is being installed in Metro São Paulo, Brazil



# Platform screen doors

- Platform Screen Doors, aka PSD, is a door system that is installed in the platform stations to avoid people to fall down to the track.
- The safety related issues are even higher than for the train door system.
- This kind of system is being installed in Metro São Paulo, Brazil
- It's being developed by Korean company



# Platform screen doors

- Besides safety constraints there is no room to rework

# Platform screen doors

- Besides safety constraints there is no room to rework
- there will be only few days for test

# Platform screen doors

- Besides safety constraints there is no room to rework
- there will be only few days for test
- The IEC 62279[14] should be followed as a documentation guide

# Platform screen doors

- Besides safety constraints there is no room to rework
- there will be only few days for test
- The IEC 62279[14] should be followed as a documentation guide
- It is requested by the standard that a formal method should be used from the detailed specification to the unit tests



# Platform screen doors

- Another problem that was faced is the lack of knowledge on formal methods and development process by the team.



# Platform screen doors

- Another problem that was faced is the lack of knowledge on formal methods and development process by the team.
- SCADE tool was selected to help on these tasks.





# Platform screen doors

- Another problem that was faced is the lack of knowledge on formal methods and development process by the team.
- SCADE tool was selected to help on these tasks.
- SCADE seems to be the best choice for non-formal method people. (based on the previous comparison)



# Platform screen doors

- Another problem that was faced is the lack of knowledge on formal methods and development process by the team.
- SCADE tool was selected to help on these tasks.
- SCADE seems to be the best choice for non-formal method people. (based on the previous comparison)
- more than 50% of the documentation and tests can be generated/performed by SCADE



# Platform screen doors

As an example, two functions that should be modeled, based on the first requirement specification of one of the PSD system equipments are shown.

- The two extractions from the Software Requirement Specification are as follows:
  - *open command* If PCM is enabled, and the OPEN button is pressed longer than 1 second, the OPEN command has to be generated.
  - *close command* If PCM is enabled, and the CLOSE button is pressed longer than 1 second, the CLOSE command has to be generated.



# Platform screen doors

Using SCADE, it was modeled like:

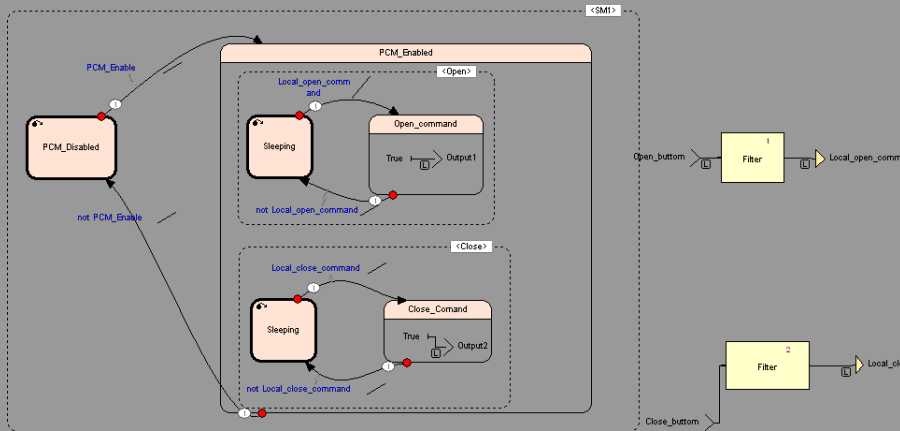


Figure: SCADE model

# Platform screen doors

- There are a lot of missing information and contradictions.



# Platform screen doors

- There are a lot of missing information and contradictions.
- It is not possible to say whether it is correct or not, to generate a close command while the open command is present, and vice versa.



# Platform screen doors

- There are a lot of missing information and contradictions.
- It is not possible to say whether it is correct or not, to generate a close command while the open command is present, and vice versa.
- There is no information needed to determine when the command (doesn't matter open or close) should be turned off.



# Platform screen doors

The main objective here was to present that a simple way to formalize the development process, whether or not, with heavy formal methods, helps a lot to find this kind of problems. It's an ongoing project, and the author hopes to present some strong evidences to support these assumptions.





# considerations

- It was not necessary to have someone with strong knowledge in mathematics, although the basic concepts were needed.



# considerations

- It was not necessary to have someone with strong knowledge in mathematics, although the basic concepts were needed.
- It was not necessary a big team in none of the described projects in order to successfully carry on the project.



## considerations

- It was not necessary to have someone with strong knowledge in mathematics, although the basic concepts were needed.
- It was not necessary a big team in none of the described projects in order to successfully carry on the project.
- The most difficult task was the requirement elicitation and analysis.



# considerations

- It was not necessary to have someone with strong knowledge in mathematics, although the basic concepts were needed.
- It was not necessary a big team in none of the described projects in order to successfully carry on the project.
- The most difficult task was the requirement elicitation and analysis.
- Formal model helps during the classification and elaboration of each requirement forcing them to be complete and non ambiguous.



# considerations

- The time (and money) that is spent in the earlier phases of the development process is greater than in a normal development



# considerations

- The time (and money) that is spent in the earlier phases of the development process is greater than in a normal development
- The time (and much money) that is spent in tests and rework is definitely less.

## Gaps or needs

- *Requirements*, It is a fact that requirement problems are responsible for more than 40% of the total problems in a project 4. Then, this is the most important feature that should be integrated to RODIN platform.

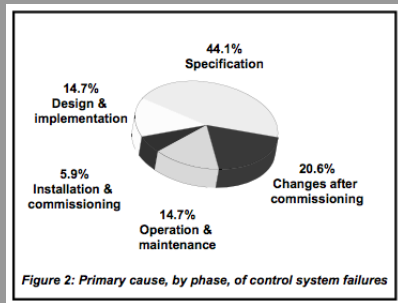


Figure: Requirement problems from [15]

# Requirements

- the development of a "natural language dictionary", that would be used to rewrite the specification in a way it could be better understood.



# Requirements

- the development of a "natural language dictionary", that would be used to rewrite the specification in a way it could be better understood.
- the conversion of this redefined specification directly into the abstract model, avoiding with that the insertion of human errors

# Requirements

- the development of a "natural language dictionary", that would be used to rewrite the specification in a way it could be better understood.
- the conversion of this redefined specification directly into the abstract model, avoiding with that the insertion of human errors
- the creation of a tool based on the formal model that could write back a natural language specification. This is crucial when modeling the model manually and at the end there is the need to present it to the customer for approval.

# Requirements

- the development of a "natural language dictionary", that would be used to rewrite the specification in a way it could be better understood.
- the conversion of this redefined specification directly into the abstract model, avoiding with that the insertion of human errors
- the creation of a tool based on the formal model that could write back a natural language specification. This is crucial when modeling the model manually and at the end there is the need to present it to the customer for approval.
- the development, (better than a simple "natural language dictionary") of a methodology to annotate the requirement files allowing verifying the coverage of this requirement and helping traceability.



# Traceability

- *Traceability* Also related to requirements, RODIN platform should have the capability to:
  - to be able to track forwards, that is, when something is changed in the abstract model, it would be good if RODIN platform could point out the possible refinements that should be verified in order to meet the changes.

# Traceability

- *Traceability* Also related to requirements, RODIN platform should have the capability to:
  - to be able to track forwards, that is, when something is changed in the abstract model, it would be good if RODIN platform could point out the possible refinements that should be verified in order to meet the changes.
  - in the same way, it should be able to track backwards, and point where to verify if changes were made (intentional or not) in the refinement machines.

# Traceability

- *Traceability* Also related to requirements, RODIN platform should have the capability to:
  - to be able to track forwards, that is, when something is changed in the abstract model, it would be good if RODIN platform could point out the possible refinements that should be verified in order to meet the changes.
  - in the same way, it should be able to track backwards, and point where to verify if changes were made (intentional or not) in the refinement machines.
  - still more crucial, it would be good if RODIN platform allowed to track back and forward all the requirements and any changes could be highlighted. Moreover, with this ability, it would be possible to verify if all requirements were fulfilled or not.

# Intermediate languages

- *intermediate languages* - This has already been done by UMLB plugin, but an interesting feature seems to be missing. Besides the ability to create state machines, for example, the ability to execute these models would be gratefully appreciated. With that, it would be possible to verify if the assumptions are correct, with no need to go inside the proof obligations.
- Some research about KAOS model is being performed also, to help model the specification and then translate it into Event B models. It could be another good approach, but it's not as known by industry as UML.

# Test case generation

- *test case generation* This seems to be one of the biggest gaps in industry right now. All generated tests are based on specialist feelings, and usually, what is tested is not exactly what should be. As a result, after a long time testing the system, at the moment it is set to operate some failure occurs, and the test generation phase has to begin again in order to address that specific failure. This routine happens several times until the product can be finally released. The Proof Obligations are strongly pointed as the basic source for generating test cases that are necessary and sufficient. If those proofs are necessary and sufficient to validate the specification, why not use those proofs to generate the test case scenarios?





# Conclusion

- The application of formal methods in industry is growing, however most of the times as a result of some projects involving academia and industry, like DEPLOY project.

If these barriers could be broken, the use of formal methods would spread out really fast.



# Conclusion

- The application of formal methods in industry is growing, however most of the times as a result of some projects involving academia and industry, like DEPLOY project.
- Outside Europe, formal methods usage is still incipient

If these barriers could be broken, the use of formal methods would spread out really fast.



# Conclusion

- The application of formal methods in industry is growing, however most of the times as a result of some projects involving academia and industry, like DEPLOY project.
- Outside Europe, formal methods usage is still incipient
- More effort in showing the benefits of formal methods usage is needed.

If these barriers could be broken, the use of formal methods would spread out really fast.



# Conclusion

- The application of formal methods in industry is growing, however most of the times as a result of some projects involving academia and industry, like DEPLOY project.
- Outside Europe, formal methods usage is still incipient
- More effort in showing the benefits of formal methods usage is needed.
- It is necessary tools that do not scare the customer in a first sight.

If these barriers could be broken, the use of formal methods would spread out really fast.



# Conclusion

If the managers are open minded, and admit waiting a bit more at the beginning of the development to see real results, (light or heavy) formal methods application could be a lot cost-effective and could, at the end, decrease the costs of the whole project by decreasing the costs in test and maintenance phases.





ClearSy:  
Atelierb



Butler, M., Hallerstede, S.:  
The rodin formal modelling tool.  
[deploy-eprints.ecs.soton.ac.uk](http://deploy-eprints.ecs.soton.ac.uk)



Esterel:  
Getting started with scade.  
(Sep 2007) 1–148



CENELEC:  
Software for Railways Control and Protection Systems. EN  
50128.  
(1995)



Abrial, J.:  
The b-book: Assigning programs to meanings.  
[books.google.com](http://books.google.com) (Jan 1996)



Commission, I.E.:

IEC 61508 - Functional safety of electrical/electronic/programmable electronic safety-related systems.

International Electrotechnical Commission Standards (1998)



Abrial, J.:

Formal methods: Theory becoming practice.

Journal of Universal Computer Science (Jan 2007)



Jetley, R., Iyer, S., Jones, P.:

A formal methods approach to medical device review.






COMPUTER (Jan 2006)



Lecomte, T., Servat, T., Pouzancre, G.:

Formal methods in safety-critical railway systems.

Proc. Brazilian Symposium on Formal Methods: SMBF (Jan 2007)

-  Abrial, C., Voisin, L.:  
Event-b language
-  Halbwachs, N., Caspi, P., Raymond, P., Pilaud, D.:  
The synchronous data flow language lustre.  
Proceedings of the IEEE **79**(9) (1991) 1304–1320
-  Harel. . . , D.:  
Statecharts: A visual formalism for complex systems.  
Science of Computer Programming (Jan 1987)
-  Berry, G.:  
The foundations of esterel.  
Foundations Of Computing Series (2000) 425–454
-  Commission, I.E.:  
IEC 62279 Railway Applications Communications, Signalling  
and Processing Systems Software for Railway Control and  
Protection Systems.



## International Electrotechnical Commission Standards (2002)



Bell, R.:

**Introduction to iec 61508.**

Proceedings of the 10th Australian workshop on Safety ...

(Jan 2006)