DA Associate Program - AeS Proposal Ongoing development

Aryldo G Russo Jr.

AeS Group & Research Institute of State of São Paulo (IPT),
agrj@aes.com.br

Abstract. I would like to present, by this proposal, our intention to become a Associate Partner of DEPLOY Project, and I state here our agreement with what is written in the DA agreement. Our plan is to disseminate the B Method and related tools, namely, RODIN Platform in South America, and to that we intend to apply the formalism and tools in our current projects related to railway field. In order to do that, we intend to start with a small project, namely "Dead man control" to prove the concept and teach all people involved, and after that, in case of good results make the formal development a standard for all projects. In this proposal I present in more detailed way an introduction of AeS company, the AeS proposal and pilot project and the expected results.

1 Introduction

1.1 The AeS Group

AeS Group is a Brazilian company and was created in 1991, and at that time it was working in the building automation field. By the year of 1998, it began his involvement in the railway field, when the first Brazilian Door Control System for Rolling stock doors were developed.

Due to the advances in technology, many safety functions that were handled by hardware are now responsibility of the embedded software. This fact triggered motivation to use formal methods in standards relevant to software safety [1]. Some standards can be followed to increase the equipment safety level. One of them is the IEC 61508 [2]. This standard presents four levels of safety, the so called Safety Integrity Levels - SIL, and above level 2, the use of formal method is required or suggested to achieve a certain level of completeness, robustness, and safety, that grows as the level grows. The goal of using formal methods is to produce an unambiguous and consistent specification that is as complete, error-free and with less contradictions as possible, however simple to verify.

To address the group concern with safety, the 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[3]).

Based on these previous information, and the constraints such as, the size of the company (at that date, AeS counted only with 15 employees, and most of them working on administrative tasks) and the lack of deep knowledge of the method itself, the AeS group decided, first, to study and use the B method[4] 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).

From that time, and after facing several pitfalls, AeS Group has acquired a reputation as a company that has the needed know-how to develop safety critical applications, and, nowadays, it is in charge of several training courses around the world teaching software development process for safety critical applications based on a formal method mind.

2 Proposal

As stated before, AeS has been involved with formal methods for several years and from 2007 was decided to go further in this field.

After several congress, seminars and workshops participations, we were invited to submit a proposal to participate more officially in the DEPLOY project.

The motivation of this work is basically the same as stated in DEPLOY document[5] for the Workpackage 2, where it's said:

> "Increasingly rail transportation systems use high degrees of software automation for both safety and control. The main concern of Siemens Transportation in DEPLOY is demonstrating the safety of the rail systems that they develop, in particular the safety of products that are reused in different operational contexts. Modifying a safe pre-existing system can lead to many unsafe scenarios that are hard to detect, especially at a system level. We believe that refinement-based formal engineering methods will enable us to manage the complexity of ensuring system safety and maintaining safety during system modification and customisation. These methods will also provide us with the evidence to demonstrate safety to a high level of assurance."

In our case, we deal with some railway sub-systems that are also safety-related or safety critical, such as, Door Control Systems, Brake Control Systems, Speed Detection Systems, etc...

Our proposal for that invitation is the creation of a structured development process based on formal methods, and in order to achieve that, we would like to try to extract from each workpackage development tasks, tools, techniques and methodologies that could better fit in each phase of the development process. This is not exactly a new proposal, as can be seen at [6], but the real implementation of this approach is not realised yet.

Moreover, this structured development process must:

- be able to be used by small companies but with the possibility to scale for the bigger ones;
- be cost effective in terms to, at least, not increase the development costs;

– be adherent to the current standards in the railway field;
– and be able to be used by people with no strong mathematical knowledge

2.1 pilot project

In order to validate this proposal, we propose a pilot project based on a small system that's used to stop the train when the operator is not possible any more (by different reasons) to apply protection actions.

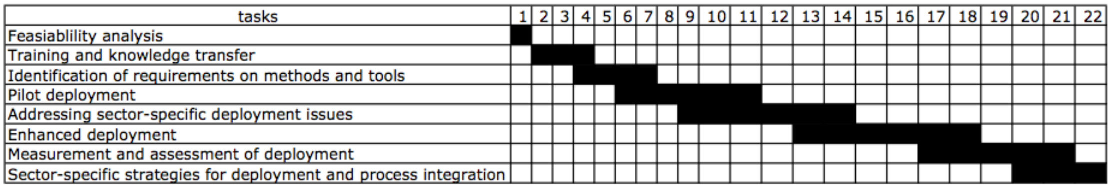
This system is called "dead man control" and the basic requirements are: (the full requirements elicitation is also part of this research)

1. if the train is in automatic mode, the dead man control must be disabled
2. if the train is in manual mode, the dead man control must be enabled
3. in manual mode the operator must push the control button each X seconds
4. after X seconds, if the operator have not pressed the button the system must provide an alarm sound
5. after X seconds after the alarm activation, if the operator has not pressed the button the system must stop the train
6. in any time the operator press the button, the system must go back to the normal situation.

To achieve the desired development process, we intend to follow the "V development model" presented in the IEC 61508 standard[2] and its derivatives, and for each phase of this process we intend to identify what should best fit among the techniques provided by DEPLOY project.

to cite some, we intend to use the requirement methodology, the safety case techniques, the decomposition, the validation. code generation, among others.

Based on the adopted strategy on DEPLOY project [5], we propose the following plan to be addressed:



tasks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Feasiablility analysis	■																					
Training and knowledge transfer		■	■	■																		
Identification of requirements on methods and tools				■	■	■	■															
Pilot deployment						■	■	■	■	■	■											
Addressing sector-specific deployment issues									■	■	■	■	■	■								
Enhanced deployment													■	■	■	■	■	■				
Measurement and assessment of deployment																	■	■	■	■	■	
Sector-specific strategies for deployment and process integration																				■	■	■

Fig. 1. proposed schedule

2.2 Project generalities

As can be seen in figure 2, the development process is composed by several phases and each phase composed by several tasks. This model was extracted from the IEC 62279 standard, although it's similar in several different fields of application.

Most of the time, when this model is followed, in safety critical applications, formal methods are used only in a small part of the process. Our objective here is to spread formal method utilisation in almost all the phases in this process, and ultimately, create a guide of application that could be used by others in order to introduce this extended methodology.

For that, we intend to use what's been developed inside DEPLOY workpackages (see figure 2 for details) to create a chain of application.

We expect to be able to integrate what's been developed in Workpackage 1 and 4 during the system and software specification phase. The effort here will be in determine a way to specify better requirements to avoid that errors like ambiguities and inconsistencies move forward to later phases. There are also other studies we'd like to investigate in this phase like in [7] and [8]

During the software architecture and design we intend to use what's been developed in workpackage 2 and 3 along with the decomposition technique that's on development, trying with that, in the determined point separate hardware part and software part, and from that continue with the next refinements independently from each other.

After code generation phase, that we hope it would be possible to be done automatically from the refined specification, we intend to use what's been developed in workpackage 9, in the sense of helping the tool development group in create an appropriate plug-in to extract test cases from the formal specification.

At the end, the whole process is part of a more wide objective that is improve the dependability of the developed product, which meets the expectation of the workpackage 8.

2.3 Project details

The main objective of this project is to create a useful methodology to be used during the development life cycle of safety critical systems. In order to do that, is a fact that some ingredients are needed, as follow:

1. A development life cycle *framework*. This framework must define what are the phases in this life cycle, and what are the inputs and outputs of these phases. Moreover, it needs to state what are the tasks that need to be performed to "transform" the inputs in the correspondent outputs of each phase. There are several frameworks that could be used, like spiral, clean room, XP, etc.. but, as it is the case of railway domain, the V model would be the one used in this project.
2. For the "transformation", or to perform each task, it's necessary some *techniques* (or languages, tools, etc...) that would be used to get the inputs and generate the expected outputs. As the objective of this project is the application of formal methods (in our understating, formal methods are, in fact, formal languages by the fact of lack of a utilization method, like is stated in

[9]) during the development life cycle, one of the expected results is the identification of what method and related tools would be suitable. As in some other previous studies, like [], we have strong feelings that the B method and its derivatives would suit well in most of the cases in railway domain. But, where is the case of necessity other formalisms would be applied, and as a secondary objective in this project we would like to evaluate, compare, and verify other methods like VDM [], Z [], and others.

3. But, to be able to use such techniques, an utilization (or application) method need to be used in order to guide, or to state the steps that are necessary to successfully achieve the objectives. In almost all cases, such formal languages are not followed with this methods, and, as was stated by Jens Bendisposto and Michael Leuschel, during Dagstuhl Seminar(Refinement Based Methods for the Construction of Dependable Systems), using the example of the "Abrial index", where can be seen that when these formal languages are used by people that really knows about that quite well, the resulting specification is easily proved where is not the case when it's done by people who not follow a (hidden) method. During this project, as another secondary objective, we would like to determine a method that would guide these techniques application, to help people during the development life cycle to spend their time in valuable tasks, and not in "try and error" experiences.
4. Finally, as the *methodology* itself, is the task to determine the transitions from one phase to another. It needs to be a guide that state what intermediate tasks are needed in order to an output of a previous phase could be used as input of the next one. Moreover, it's the translation of the used framework in words that state how to perform whatever is needed to achieve the end of the life cycle, and not only the "what" needs to be done. This is the main objective of this project, and we hope it could be generic enough that could be used in other domains, but strong enough to facilitate the adoption of formal methods in railway domain as a strong methodology that helps the accomplishment of what is already required by the domain standards.

To clarify what we intend to perform, we present in the next section our first assumptions about what might be applied in each steps of the framework presented before.

2.4 assumptions

As can be seen in figure 2, there are some phases that are required by the framework we decide to adopt, namely[1]:

1. System Development Phase (Table 1)
2. Software Requirements Specification Phase (Table 2)
3. Software Architecture and Design Phase (Table 3)

[1] One thing that need to be noted is that, from phase 2 until phase 9, the same principle might be applied to hardware development as well, and this is another field of interest of this project, but it will be discussing later in this article.

4. Software Module Design Phase (Table 4)
5. Code Phase (Table 5)
6. Software Module Testing Phase (Table 6)
7. Software integration Phase (Table 7)
8. Software/Hardware integration Phase (Table 8)
9. Software Validation Phase (Table 9)
10. Software Assessment Phase
11. Software Maintenance Phase

What was done in this very early stage of the projects, were some assumptions about what we could do in each phase of this life cycle, related to what technique we might use, how would be the method for this usage and in a overall view, what would be the resulting methodology we should apply during the life cycle to support the phase transitions.

we present bellow these assumptions in a table form, meaning that for each phase we present a table stating the tasks for that phase and the corresponding technique the we suppose would fit well for that propose.

TASKS	**TECHNIQUE ASSUMPTION**
System Requirement Specification	Problem Frames
System Safety Requirements Specification	Safety analysis integration
System Architecture Description	??? Problem Frames

Table 1. System Development Phase

Method here it's necessary to describe which method would be used.

TASK	**TECHNIQUE ASSUMPTION**
Software Requirement Specification	????? Use Case approach
Software Requirements Test Specification	????

Table 2. Software Requirement Specification Phase

Method here it's necessary to describe which method would be used.

Method here it's necessary to describe which method would be used.

Method here it's necessary to describe which method would be used.

Method here it's necessary to describe which method would be used.

TASK	TECHNIQUE ASSUMPTION
Software Architecture Specification	?????
Software Design Specification	??? formal model

Table 3. Software Architecture and Design Phase

TASK	TECHNIQUE ASSUMPTION
Software Module Design Specification	refined and decomposed formal model
Software Module Test Specification	proofs

Table 4. Software Module Design Phase

TASK	TECHNIQUE ASSUMPTION
Software Source Code	generated from formal model
Software Source Code verification	proofs

Table 5. Code Design Phase

TASK	TECHNIQUE ASSUMPTION
Software Module Testing report	????

Table 6. Software Module Testing Phase

Method here it's necessary to describe which method would be used.

Method here it's necessary to describe which method would be used.

Method here it's necessary to describe which method would be used.

Method here it's necessary to describe which method would be used.

3 Metrics

In order to verify the convenience of this methodology, some metrics need to be defined in order to evaluate some points related to the normal development process. this points and metrics are listed bellow.

4 Expect results

As the main result of this research we hope we can build a development guide where the several different techniques will be presented and how each of the chosen techniques must be applied in each development phase in order to build a safe system based on a formal approach.

TASK	TECHNIQUE ASSUMPTION
Software Integration Test report	????

Table 7. Software Integration Phase

TASK	TECHNIQUE ASSUMPTION
Software/Hardware Integration Test report	????

Table 8. Software/Hardware Integration Phase

As a side effect we expect to be able to determine what should be the best way to introduce or teach industrial partners on the use of this kind of formalisation.

Moreover, we expect that based on this development process, the resulting system will be more easily modifiable, customisable and and the safety product itself more reusable.

If the concept presented here can be proved as a convenient way to develop safety related products, we intend to implement this methodologies in our current development process, and also, through our consultant services, introduce this in other companies.

5 Remarks

Besides safety related issues, there is a growing need for deploying formal methods in specification of security properties in real industrial applications, what can be in the future another field of application of this methodology

[10,11,12,13,14,15]

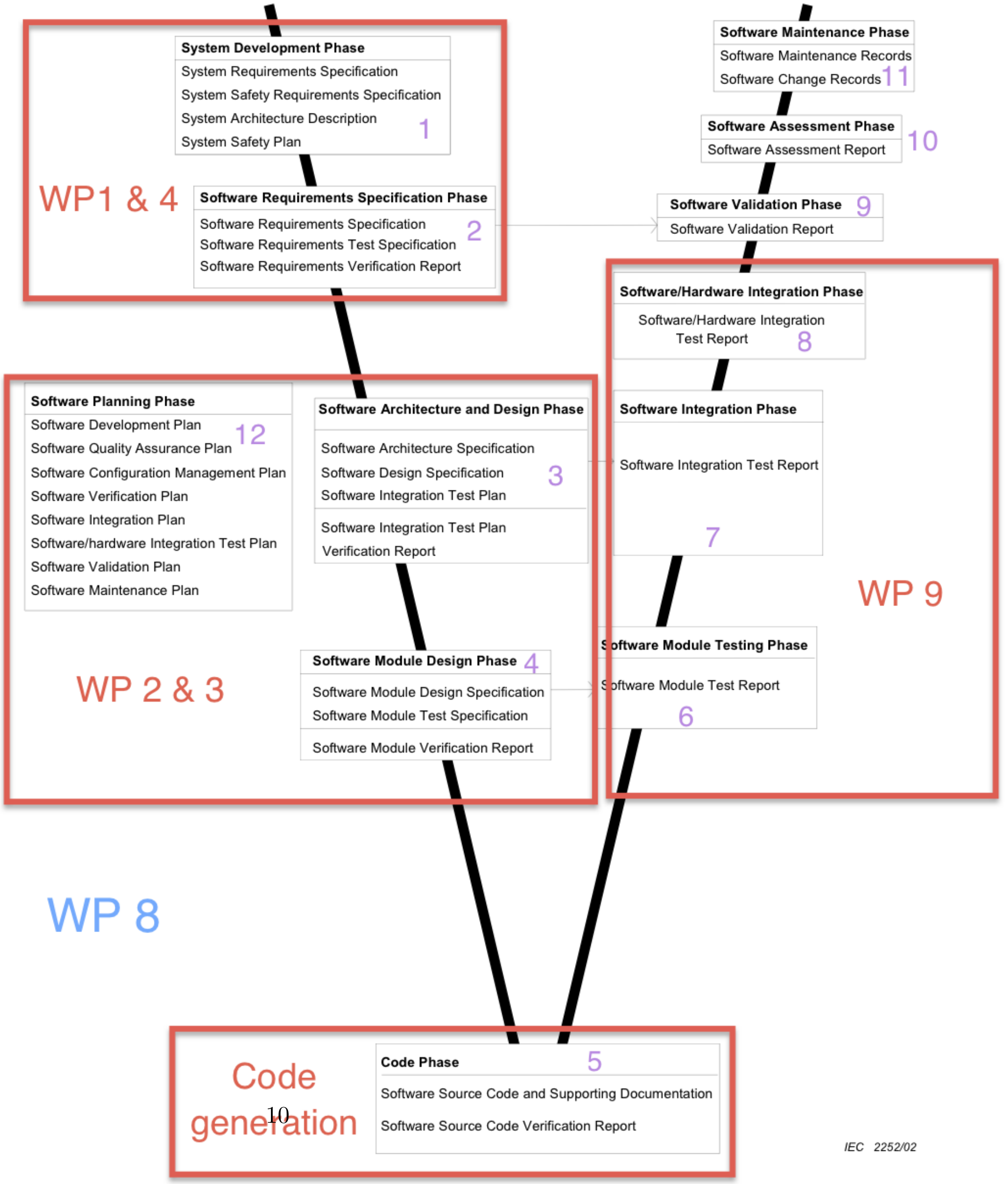
References

1. Bowen, J.P., Stavridou, V.: The industrial take-up of formal methods in safety-critical and other areas: A perspective. In: FME '93: Industrial-Strength Formal Methods, First International Symposium of Formal Methods Europe. Volume 670 of Lecture Notes in Computer Science., Odense, Denmark, Springer (1993) 183–195 1
2. Commission, I.E.: IEC 61508 - Functional safety of electrical/electronic/programmable electronic safety-related systems. International Electrotechnical Commission Standards (1998) 1, 3
3. CENELEC: Software for Railways Control and Protection Systems. EN 50128. (1995) 1
4. Abrial, J.: The b-book: Assigning programs to meanings. books.google.com (Jan 1996) 2
5. Tochtermann, K., Granitzer, G., Pillmann, W., Geiger, W.: Ict-ensure–a 7 th framework program support action for building the european research area in the field of ict for environmental sustainability. Environmental Informatics and Industrial Ecology, Proc. of the EnviroInfo (2008) 10–12 2, 3

TASK	TECHNIQUE ASSUMPTION
Software Validation report	????

Table 9. Software Validation Phase

6. Hall, A.: Realising the benefits of formal methods, formal methods and software engineering, lncs 3785. (2005) 2
7. Lamsweerde, v., A: Goal-oriented requirements engineering: A guided tour. Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on (2001) 249–262 4
8. Gunter, C.A., Gunter, E.L., Jackson, M., Zave, P.: A reference model for requirements and specifications. IEEE SOFTWARE **17**(3) (2000) 37–43 4
9. Mazzara, M.: Deriving specifications of dependable systems: toward a method. In: Proceedings of the 12th European Workshop on Dependable Computing (EWDC 2009). (2009) 5
10. Aryldo, J.G.R.: Formal methods in industry: The state of practice of formal methods in south america and far east. Dagstuhl Seminar (2009) 8
11. Jastram, M., Leuschel, M., Bendisposto, J., Jr, A.G.R.: Mapping requirements to b models. DEPLOY Deliverable (5 2009) 1–19 8
12. Jr, A.G.R., Jr, N.B.: Modelamento formal dos processo de troca de documentos nos protocolos e-commerce. (12 2008) 8
13. eharbe, D.D., Moreira, A.M., Silva, P.S.M., Jr, A.G.R.: Modelling control systems in b: an industrial case study. SBMF 2007 - 10th Symposium on Formal Methods (7 2007) 16 8
14. Russo Jr, A.G.R.J., de Sousa, T.C.: Starting b specifications from use cases. (2009) 8
15. Jr, A.G.R., Jr, N.B.: Uma proposta para elicitação de requisitos não-funcionais. SULCOMP2008 (9 2008) 1–10 8


System Development Phase
System Requirements Specification
System Safety Requirements Specification
System Architecture Description
System Safety Plan
1
WP1 & 4
Software Requirements Specification Phase
Software Requirements Specification
Software Requirements Test Specification
Software Requirements Verification Report
2
Software Maintenance Phase
Software Maintenance Records
Software Change Records
11
Software Assessment Phase
Software Assessment Report
10
Software Validation Phase
Software Validation Report
9
Software/Hardware Integration Phase
Software/Hardware Integration Test Report
8
Software Planning Phase
Software Development Plan
Software Quality Assurance Plan
Software Configuration Management Plan
Software Verification Plan
Software Integration Plan
Software/hardware Integration Test Plan
Software Validation Plan
Software Maintenance Plan
12
Software Architecture and Design Phase
Software Architecture Specification
Software Design Specification
Software Integration Test Plan
Software Integration Test Plan Verification Report
3
Software Integration Phase
Software Integration Test Report
7
WP 9
Software Module Design Phase
Software Module Design Specification
Software Module Test Specification
Software Module Verification Report
4
WP 2 & 3
Software Module Testing Phase
Software Module Test Report
6
WP 8
Code generation
Code Phase
Software Source Code and Supporting Documentation
Software Source Code Verification Report
5

IEC 2252/02

Figure 4 – Development Life Cycle 2