The 4SECURail case study on rigorous standard interface specifications

D. Belli[1], A. Fantechi[2], S. Gnesi[1], L. Masullo[3], F. Mazzanti[1],
L. Quadrini,[3] D. Trentini[3], and C. Vaghi[4]

[1] ISTI-CNR, Via G. Moruzzi 1, Pisa 56124, Italy
[2] DINFO, University of Florence, Via S. Marta 3, Firenze, Italy
[3] MER MEC STE, via Bombrini 11, Genova 16149, Italy
[4] FIT Consulting, Via Sardegna 38 Roma – 00157, Italy

Abstract. In the context of the Shift2Rail open call S2R-OC-IP2-01-2019, one of the two work streams of the 4SECURail project has pursued the objective to corroborate how a clear, rigorous standard interface specification between signaling sub-systems can be designed by applying an approach based on semi-formal and formal methods. The objective is addressed by developing a demonstrator case study of the application of formal methods to the specification of standard interfaces, aimed at illustrating some usable state-of-the-art techniques for rigorous standard interface specification, as well as at supporting a Cost-Benefit Analysis to back this strategy with sound economic arguments.

1 Introduction

In an increasingly competitive market as the railway one, the application of Formal Methods (FM) within the process of developing standard interfaces between signaling sub-systems is believed to be a winning strategy for the construction of high-quality, safe, and reliable signaling infrastructure, gaining in this way the interest by the infrastructure managers (IMs). Such a trend is fostered by economic and technical reasons. Economic reasons can be found, besides the market competition, in the reduction of both vendor lock-in effect and costs caused by change requests due to requirements inconsistencies. Technical reasons concern the reduction of interoperability problems and the fact that clear, rigorous specifications of standard interfaces are well suited to exploit formal methods within the development of signaling systems.

In the context of the Shift2Rail open call S2R-OC-IP2-01-2019, one of the two work streams of the 4SECURail project [24] has pursued the objective to corroborate how a clear, rigorous standard interface specification can be designed by applying an approach based on semi-formal and formal methods.

The work stream was intended at defining (and prototyping) a demonstrator of the use of state-of-the-art formal methods for the development and analysis of standard interfaces, with measured cost/benefit ratio for the industrial application of the demonstrated process. The activity of the project included hence i) the specification of the demonstrator for the use of formal methods in railway signalling by identifying, selecting, and composing appropriate formal methods and tools for industrial application; ii) the identification of a railway signaling system to be used as a test case, composed of subsystems that should interoperate by means of standard interfaces, to exercise the formal methods demonstrator, iii) use the developed test case as an information source to base a Cost-Benefit Analysis of formal methods usage in the railway signalling domain.

In this paper we show the main results of the relevant workstream of the 4SECURail project, both in terms of recommended techniques for the specification of standard interfaces (Sect. 2) and of a Cost-Benefit Analysis of the adoption of formal methods in the railway industry (Sect.3). Section 4 draws some conclusions.

2 The demonstrator case study

The current trend in the direction of clear and rigorous specifications of standard interfaces is to complement the use of natural language requirements with graphical SysML/UML artifacts - see, e.g., EULYNX [7]. However, the unrestricted use of SysML/UML as a specification language for "Systems of Systems" (SoS) can be problematic because of its genericity and the lack of precise semantics. SysML/UML conceals many hidden assumptions that may have a strong impact on the expected behaviors of the modeled system. Formal models that can be rigorously analysed need instead to be mechanically associated with the semi-formal SysML/UML-based designs. The goal of our work is to show a possible approach and highlight the pros and cons of the application of formal methods for the specifications of standard interfaces.

The adopted methodology is fully described in Deliverables D2.1 [18], D2.2 [17], and D2.5 [19]) of the 4SECURail project, and is exemplified with the development of a demonstrator that illustrates the application of formal methods to a selected case study. Deliverable D2.3 [22] describes the details and rationale of the selected case study, which is based on the RBC/RBC communication layer that supports the execution of the RBC/RBC handover protocol.

2.1 The 4SECURail case study

The European Train Control System (ETCS) acts as an automatic train protection system which continuously supervises the train movements on a line to ensure their safe speed and distancing. To this purpose, a Radio Block Centre (RBC) communicates with all the trains in its supervision area. The transit of a train from an area supervised by a Radio Block Centre (RBC) to an adjacent area supervised by another RBC occurs during the so-called RBC-RBC handover phase and requires the exchange of information between the two RBCs according to a specific protocol. This exchange of information is supported by the communication layer specified within the documents: UNISIG SUBSET-039 [25], UNISIG SUBSET-098 [26], and UNISIG SUBSET-037 [27], and the whole stack

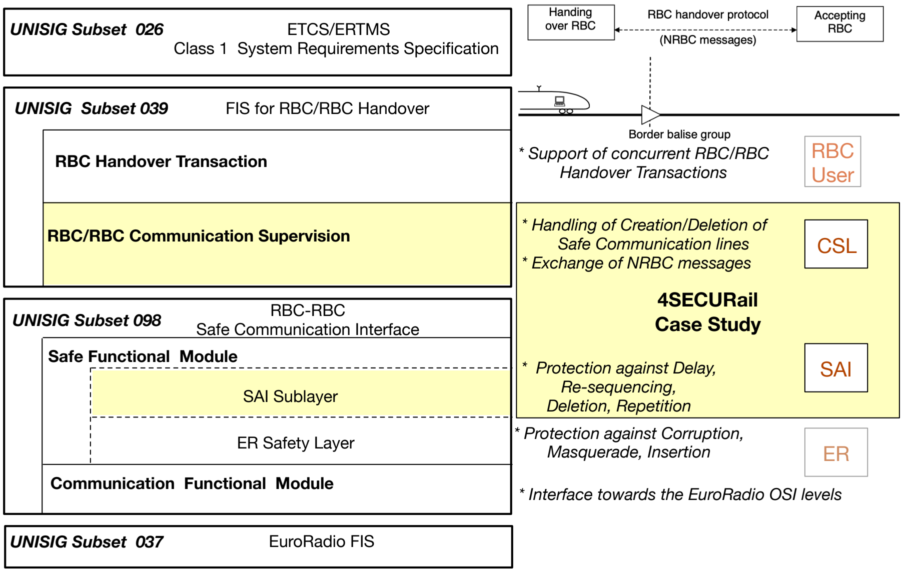

UNISIG Subset 026
ETCS/ERTMS
Class 1 System Requirements Specification
UNISIG Subset 039
FIS for RBC/RBC Handover
RBC Handover Transaction
RBC/RBC Communication Supervision
UNISIG Subset 098
RBC-RBC
Safe Communication Interface
Safe Functional Module
SAI Sublayer
ER Safety Layer
Communication Functional Module
UNISIG Subset 037
EuroRadio FIS
Handing over RBC
RBC handover protocol
(NRBC messages)
Accepting RBC
Border balise group
* Support of concurrent RBC/RBC Handover Transactions
RBC User
* Handling of Creation/Deletion of Safe Communication lines
* Exchange of NRBC messages
CSL
4SECURail Case Study
* Protection against Delay, Re-sequencing, Deletion, Repetition
SAI
* Protection against Corruption, Masquerade, Insertion
ER
* Interface towards the EuroRadio OSI levels

Fig. 1: Overall structure of the 4SECURail case study

is implemented by both sides of the communication channel. Figure 1 summarizes the overall relation between the components of the UNISIG standards, supporting the handover of a train. The components considered in the case study are the Communication Supervision Layer (CSL) of the SUBSET-039 and the Safe Application Intermediate SubLayer (SAI) of the SUBSET-098. These two components are the main actors that support the creation/deletion of safe communications and protect the transmission of messages exchanged. In particular, the CSL is responsible for requesting the activation – and in the event of failure, the re-establishment – of the communications, for keeping controlling its liveliness, and for the forwarding of the handover transaction messages. The SAI is responsible for ensuring that there are no excessive delays, repetitions, losses, or reordering of messages during transmission. This is achieved by adding sequence numbers and time-related information to the RBC messages. The RBC/RBC communication system consists of two sides that are properly configured as "initiator" and "called".

With respect to the SUBSET-098, the 4SECURail case study neither includes the EuroRadio Safety Layer (ER), which is responsible for preventing corruption, masquerading and insertion issues during the communications, nor the lower Communication Functional Module (CFM) interface. With respect to the SUBSET-039, the 4SECURail case study does not include the description of the activation of multiple, concurrent RBC-RBC handover transactions when trains move from a zone supervised by an RBC to an adjacent zone supervised by another RBC. From the point of view of the CSL, the RBC messages are forwarded to/from the other RBC side without the knowledge of their specific contents or the session to which they belong.

2.2 The formalization of the case study

The goal is to demonstrate how formal methods provide an even more efficient requirements definition, reducing development problems related to residual uncertainties, and improving interoperability of different implementations.

The overall approach followed during the modeling and analysis process is incremental and iterative. About 53 versions of the system have been generated, each one widening the set of requirements of the case study modeled, and each one passing through the steps of semi-formal and formal modeling and analysis. During this iterative process, four kinds of artifacts have been generated and kept aligned:

1. An abstract, semi-formal UML state machine design of the components under analysis.
2. A more detailed executable version of the same UML state machines.
3. A set of formal models derived from the executable UML state machines.
4. A natural language rewriting of the requirements based on the designed and analysed models.

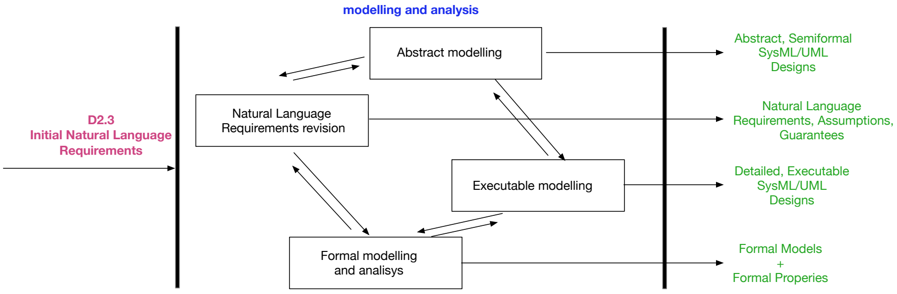

modelling and analysis
Abstract modelling
Natural Language Requirements revision
Executable modelling
Formal modelling and analisys
D2.3 Initial Natural Language Requirements
Abstract, Semiformal SysML/UML Designs
Natural Language Requirements, Assumptions, Guarantees
Detailed, Executable SysML/UML Designs
Formal Models + Formal Properies

Fig. 2: The 4SECURail demonstrator generated artifacts

Figure 2 depicts the relationship between these artifacts. The activity of generating and elaborating most of the shown artifacts (currently) requires a human problem understanding and solving activity, apart from the generation of the formal models starting from the UML executable ones, that can be (and has been in part) automated.

The natural language requirements describe the system at a high abstraction level, omitting all the details related to irrelevant implementation issues. On the contrary, during the executable modeling, which is the base for formal modeling and analysis, we need to specify these details as well.

In fact, we found it useful to introduce an intermediate level of ”abstract modeling” where the logical structure, interfaces, and the expected main control flow of the system are modeled in a rigorous notation, while irrelevant implementation issues are still described in an abstract way using natural language. These abstract models need to be further refined into executable models prior to the formal modeling activity.

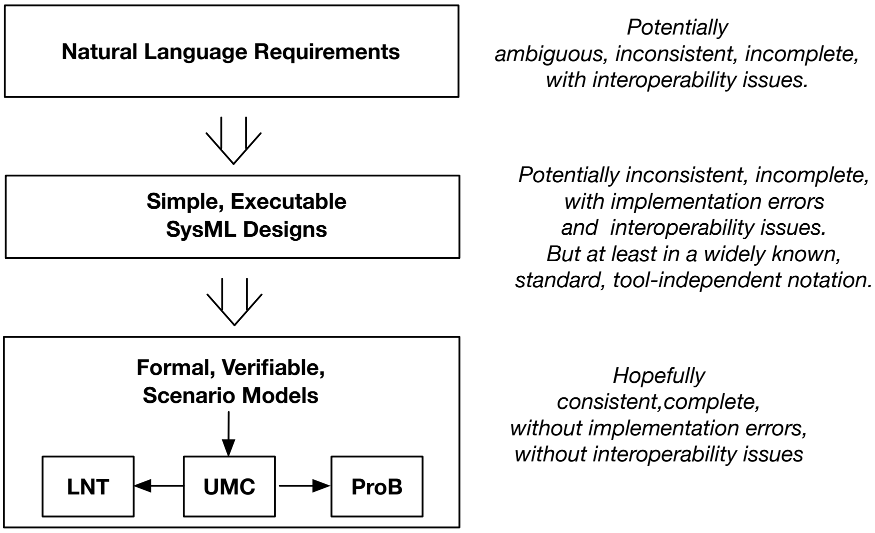

Natural Language Requirements
Potentially
ambiguous, inconsistent, incomplete,
with interoperability issues.
Simple, Executable
SysML Designs
Potentially inconsistent, incomplete,
with implementation errors
and interoperability issues.
But at least in a widely known,
standard, tool-independent notation.
Formal, Verifiable,
Scenario Models
LNT
UMC
ProB
Hopefully
consistent,complete,
without implementation errors,
without interoperability issues

Fig. 3: From natural language to formal models

As a first formal modeling step, the executable UML system diagrams corresponding to a given scenario are translated into the notation accepted by the UMC tool[5], chosen as the target of the initial formal encoding because it is a tool natively oriented to fast prototyping of SysML systems. At the beginning of the project, the possibility of designing the SysML system using a commercial MBSE framework – namely SPARX-EA[6] – has been evaluated. But implementing a translator from the SPARX-generated XMI towards UMC would have been a significant effort and it would have tied the whole analysis approach to a specific commercial tool, a fact which was not considered desirable.

Therefore, our initial SysML models have the structure of simple graphical designs; their role is just to constitute an intermediate, easy-to-understand documentation halfway between the natural language requirements and the formal models. A detailed description of these SysML models is presented in [19,21,3]. The translation of the SysML designs in the UMC notation constitutes a step towards a full formalization: UMC supports a textual notation of

[5] `https://fmt.isti.cnr.it/umc`

[6] `https://sparxsystems.com/products/ea/index.html`

UML state-machine diagrams that directly reflects the graphical counterpart [7], allows fast state-space exploration, state- and event-based (on-the-fly) model checking, and detailed debugging of the system. However, UMC is essentially a teaching/research-oriented academic tool and lacks the maturity, stability, and support level required by an industry-usable framework.

So, we have planned the exploitation of other, more industry-ready, formal frameworks and further formal models have been automatically generated in the notations accepted by the ProB[8] and CADP/LNT[9] tools (Fig 3). ProB has been selected as the second target of formal encoding because of its recognized role - see [9] - in the field of formal railway-related modeling. It provides user-friendly interfaces and allows LTL/CTL model checking, state-space exploration, state-space projections, and trace descriptions in the form of sequence diagrams. CADP/LNT has been selected as the third target of the formal encoding, because of its theoretical roots in Labelled Transition Systems, that allow reasoning in terms of minimization, bisimulation, and compositional verification. CADP is a rich toolbox that supports a wide set of temporal logic and provides a powerful scripting language to automate verification runs .

There are indeed several ways in which SysML/UML designs might be encoded into the ProB and LNT formal notations. In our case, we made the choice to generate both ProB and LNT models automatically from the UMC model. The translation implemented in our demonstrator is still a preliminary version and does not exploit at best all the features potentially offered by the target framework. Nevertheless, the availability of automatic translation proved to be an essential aspect of the demonstrated approach. Our models and scenarios have been developed incrementally, with a long sequence of refinements and extensions. At every single step, we have been able to quickly perform the lightweight formal verification of interest with almost no effort. This would not have been possible without an automatic generation of the ProB and LNT models. This approach based on the exploitation of formal methods diversity allows us to take advantage of the different features provided by the different verification frameworks.

In Figure 4 we present a table reporting the main verification features supported by the three formalization frameworks, highlighting in purple those features that require more advanced knowledge of the underlying theory and tools, while the list of features colored in black represent features that do not require a specific advanced background in formal methods to be used (e.g., analysis that can be carried out by just pushing a button). Another important advantage of our ”formal methods diversity” approach is that it allows us to verify the absence of errors in the frameworks and in the translators by checking the equivalence of the formal models and the verification results. In all three frameworks,

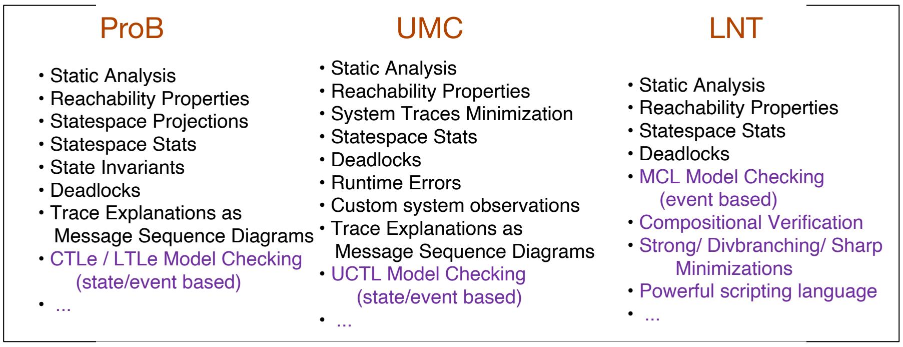
[7] actually, often it is a graphical representation that is automatically generated from the UMC encoding

[8] `https://prob.hhu.de/`

[9] `https://cadp.inria.fr/`

in fact, the underlying semantic model is a finite automaton whose transitions from state to state correspond to a single run-to-completion step of one of the state-machines that constitute the system. To show the equivalence of the UMC, ProB and LNT models we exploit the UMC feature that allows to decorate the semantic LTS of the system in a custom way, and export it in the Alderabaran ".aut" format. When comparing the ProB and UMC models, the UMC LTS is decorated with the transition labels of the UML model. These labels actually correspond to the names of the ProB "Operations" that trigger in ProB the system evolution. The LTS corresponding to the ProB evolutions is automatically generated with custom-developed translators, still in the ".aut" format, from state-space description generated by the tool itself. The two LTS can be formally proved to be strongly equivalent. When comparing the LNT models and the UMC models, the UMC LTS is this time decorated with the communication action occurring during a run-to-completion step (or with the transition label if no communication occurs). On the LNT side, the semantic LTS, which can be exported in the ".aut" format by default, is decorated with the synchronization actions of the various processes (or with an internal action identical to UMC transition label if no communication occurs). Again, the two LTS can be proved to be strongly equivalent using standard equivalence checkers working on LTS in the ".aut" format. Since the same UMC semantic model, even if differently labelled, has been proven to be strongly equivalent to the other two semantic models, we can conclude that the three models are actually equivalent. In fact, even without performing all the transformations and equivalence checking, just observing the number of states and edges of the LTS in the three models gives immediate feedback on the presence of translation errors or differences in the three execution engines.

For more details on the case study see [22], while for a detailed description of the generation process and the generated models, we refer to [19,20].



ProB

- Static Analysis
- Reachability Properties
- Statespace Projections
- Statespace Stats
- State Invariants
- Deadlocks
- Trace Explanations as Message Sequence Diagrams
- CTLe / LTLe Model Checking (state/event based)
- ...

UMC

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

LNT

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

Fig. 4: Push-button (black) and advanced features (purple) of the three adopted formalization frameworks.

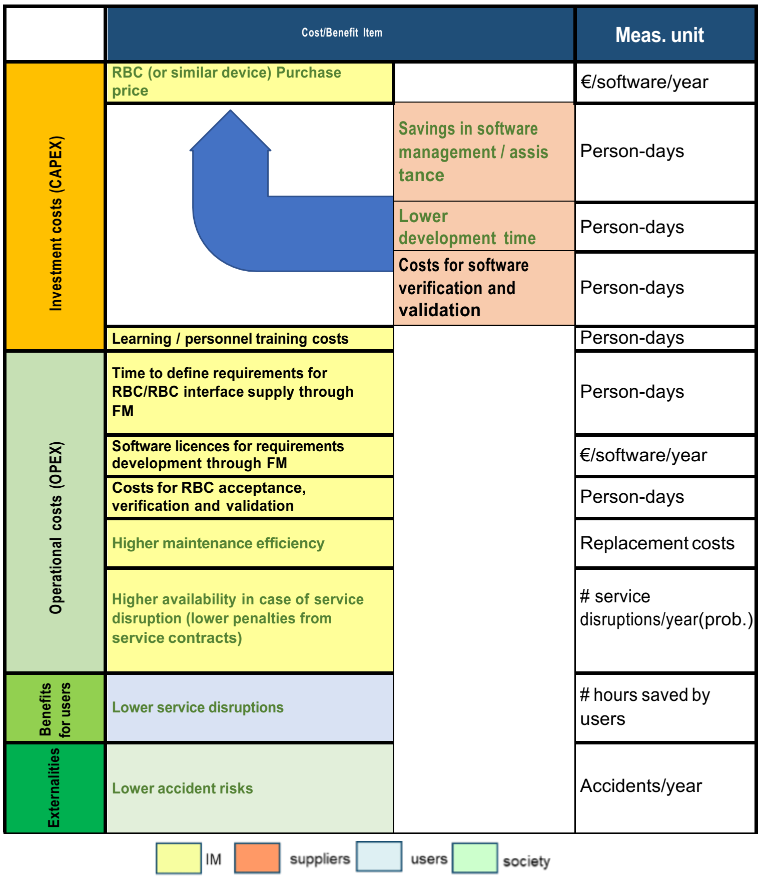
Summarizing, the demonstrator has provided explicit evidence about the advantages and difficulties associated with the introduction of formal methods in the standardization of specifications of railway systems, in particular in relation to their SoS nature. Furthermore, it has shown how the application of formal methods can provide useful feedback for improving the process of writing specifications, and how formal methods can detect and help to solve ambiguities and uncertainties introduced by natural language and semi-formal descriptions.

3 Cost-Benefit Analysis

A further objective of the 4SECURail project has been to perform a Cost-Benefit Analysis (CBA) of the adoption of formal methods in the railway industry. The final result of the analysis can be found in the 4SECURail Deliverable D2.6 [30]. We are not aware of any existing fully-fledged CBA applied to cases of formal methods adoption in the railway sector.

The 4SECURail CBA was developed taking the point of view of the Infrastructure Manager (IM), which bears the costs of the manufacturer/developer as prices. Thus, while the 4SECURail demonstrator experience has allowed us to directly observe and evaluate the potential costs of a rigorous approach in requirements specification and analysis, the quantitative evaluation of the future benefits of the approach cannot be performed by observing just the activity carried on in the project. The literature survey in [29] reports some examples of assessment of benefits of adoption of formal methods in the railway sector, but only in a limited number of cases some partially usable quantitative data are available (e.g., see [16,31,12,13,23,11,10,14,4,5,8]).
The CBA presented in this section includes two main contributions: on one side, a methodological framework to conduct the analysis is set up, with the definition of cost/benefit categories tailored to the case study (but reasonably adaptable to different formal methods application case studies) on the other hand, the instantiation of the framework is carried out by a careful assessment of actual values of cost and benefits per each category.

The adopted CBA methodology follows the guidelines set in the European Commission Guide to Cost-Benefit Analysis reported in [6], and is composed of (i) Financial Analysis, which includes the assessment of additional costs borne and additional savings accrued by an IM faced by the choice to use formal methods, and costs/benefits for suppliers, e.g., savings in terms of shorter time needed for software development, that is reflected in the price paid by IMs to purchase a RBC (of which the RBC/RBC handover interface, addressed in the 4SECURail demonstrator, is a key component); (ii) Economic Analysis, which considers benefits for users, i.e., passengers of train services, and for the "society" at large. Relevant categories of costs and benefits for the CBA have been identified (Fig. 5), such as additional costs for learning Formal Methods and for developing, by means of FM, tender specifications for the procurement of a railway signaling component, as well as savings in software development, verification, and validation, benefits for rail users due to higher maintenance efficiency, higher service availability and time saved for a lower probability of service disruption.



	Cost/Benefit Item		Meas. unit
Investment costs (CAPEX)	RBC (or similar device) Purchase price		€/software/year
		Savings in software management / assis tance	Person-days
		Lower development time	Person-days
		Costs for software verification and validation	Person-days
	Learning / personnel training costs		Person-days
Operational costs (OPEX)	Time to define requirements for RBC/RBC interface supply through FM		Person-days
	Software licences for requirements development through FM		€/software/year
	Costs for RBC acceptance, verification and validation		Person-days
	Higher maintenance efficiency		Replacement costs
	Higher availability in case of service disruption (lower penalties from service contracts)		# service disruptions/year(prob.)
Benefits for users	Lower service disruptions		# hours saved by users
Externalities	Lower accident risks		Accidents/year

🟨 IM 🟧 suppliers 🟦 users 🟩 society

Fig. 5: Cost/Benefit matrix representing the structure of relevant cost/benefit categories. In the item columns, costs are written in black and benefits in green. The respective loser/beneficiary stakeholder is indicated by the background colour, according to the bottom line.

As it is evident from Figure 5, a major part of the cost/benefit items are borne/gained by IMs (or, more properly, one single IM), from which point of view the CBA is developed. However, the scheme identifies also cost items assumed to be borne by the suppliers (i.e. by one developer, supplying one IM) and paid out by the IM through the SW purchase price. The latter is assumed to decrease with respect to the Baseline scenario, as a result of the savings accounted for by the supplier.

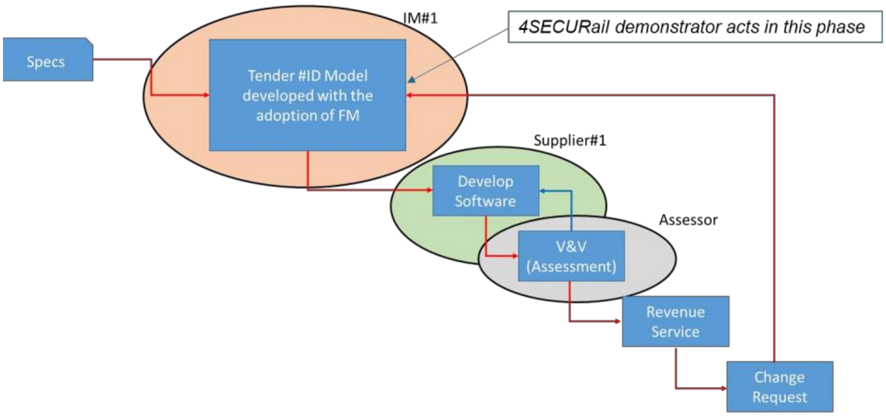
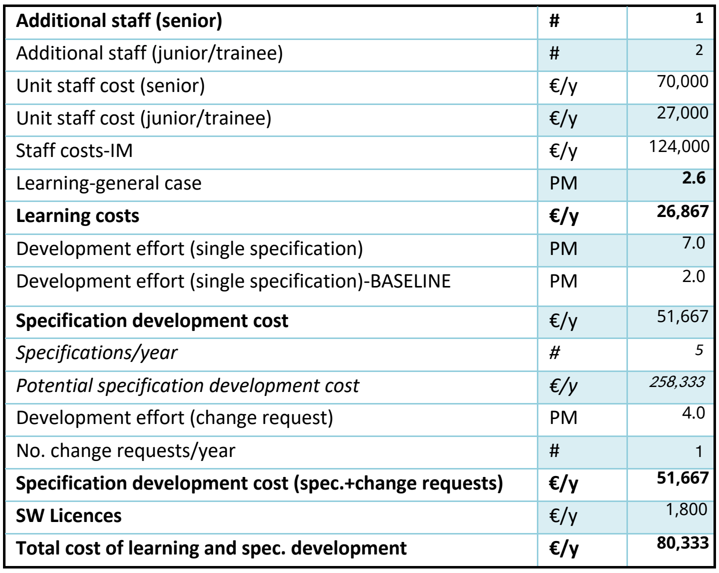

IM#1
4SECURail demonstrator acts in this phase
Specs
Tender #ID Model developed with the adoption of FM
Supplier#1
Develop Software
Assessor
V&V (Assessment)
Revenue Service
Change Request

Fig. 6: "Semi-formal methods development" business case. Each oval represents the competence area of the indicated stakeholders, i.e. the activity in which IMs, suppliers and assessors bear costs.

A micro, bottom-up case study for CBA was set-up to assess costs and savings borne by an IM faced with the choice to use FM in the development of specifications for the provision of RBC-RBC handover interfaces, vs. the baseline scenario, that is the development with no use of FM. This allowed a proper assessment of actual values of cost and benefits per each category set in Fig. 5. The business case of "semi-formal methods development" (mirroring the parallel business model proposed in the X2RAIL-2 project reported in [1]) assumes the adoption of a "tender model", in which tender requirements are developed with the use of FM (Fig. 6).

The quantitative assessment of cost and benefit categories was possible by integrating the outcome of the demonstrator developed in 4SECURail, and by assumptions based on literature and on Consortium's knowledge and experience, so overcoming the lack of fully comparable case studies, data confidentiality of software developers, and low availability of quantitative cost data about FM adoption. The assessment of effort per process and related costs made in 4SECURail [30] led to the calculation of the total cost of learning and specifications development, as borne by the IM during the 15-year time horizon. Table 1 summarises such calculations, covering all cost items assumed to be borne by the IM each year, except learning costs and software licenses, that are assumed to occur every 5 years. Unit staff costs for juniors and trainees are assumed to increase every year of a 5-year cycle, after which the staff turnover applies. As per Table 1, costs for the development of specifications (one + change requests per year) range from nearly 80,000 €/year, at the beginning of the time horizon, to 160,000 €/year at the end of the learning cycle, i.e. in the year characterized by the highest cost of juniors/trainees. Additional costs and potential savings brought by FM to the development of RBC/RBC interface (and RBC software) by IM's suppliers (i.e. software developers) are assumed to be reverted to software purchase price: if the supplier saves on development and verification and validation (V&V) costs, such savings determine a proportional decrease of the purchase price of the software ordered (through a tender process) by the IM. This assumption is once again in line with the fully-competitive perspective adopted in 4SECURail, made possible – or at least facilitated - by the adoption of FM in the development of specifications, which ultimately determines a lower dependence of an IM from a single long-term supplier. In the CBA 20% time savings were assumed for software development and V&V, leading to a financial saving of Euro 21,000 per year.



Additional staff (senior)	**#**	**1**
Additional staff (junior/trainee)	#	2
Unit staff cost (senior)	€/y	70,000
Unit staff cost (junior/trainee)	€/y	27,000
Staff costs-IM	€/y	124,000
Learning-general case	PM	**2.6**
Learning costs	**€/y**	**26,867**
Development effort (single specification)	PM	7.0
Development effort (single specification)-BASELINE	PM	2.0
Specification development cost	€/y	51,667
Specifications/year	#	*5*
Potential specification development cost	€/y	*258,333*
Development effort (change request)	PM	4.0
No. change requests/year	#	1
Specification development cost (spec.+change requests)	**€/y**	**51,667**
SW Licences	€/y	1,800
Total cost of learning and spec. development	**€/y**	**80,333**

Table 1: Learning and specification development – annual costs

The convenience for the IM to adopt FM is connected to the economies of scale generated by the replication of savings in software re-development in reply to change requests, issued by the IM through further tender processes. Since such economies of scale are likely verifiable but not easily quantifiable, the analysis has followed up with the identification – entailing a sensitivity analysis - of the optimal scale for which the additional resources deployed by the IM generate enough savings in software development to balance the additional investment and operational effort needed. In other words, the sensitivity analysis aimed at detecting what is the business scale for which the higher effort borne by the IM is balanced by savings in the development of the interface, and how much suppliers should save in the development of interfaces in reply to change requests, to ensure a competitive purchase price.

As evidenced in Figure 7, according to 4SECURail assumptions, the break-even between additional costs borne by IM and savings is verified if the purchase price of software upon change requests is -40% vs. the baseline.

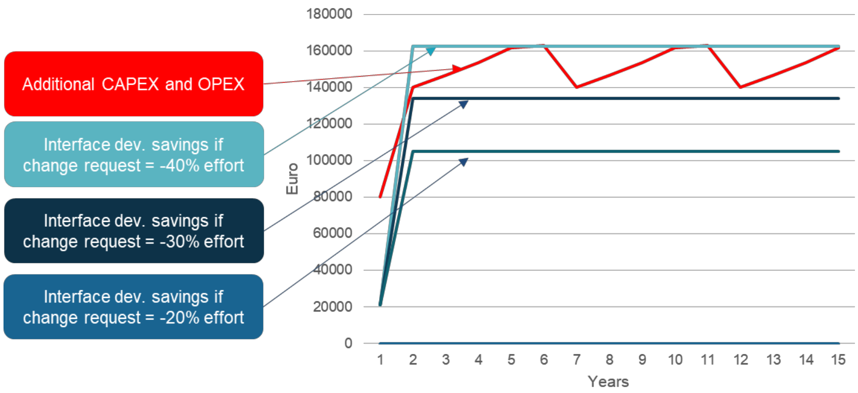


Fig. 7: Computation of break-even between additional costs and savings

In this scenario, the total annual savings for suppliers are 162,600 Euro/year, which overcome additional costs borne by IMs every year except the ending learning cycle one, i.e. when the labour cost has increased to the maximum assumed during the learning cycle. The calculation process is shown in Table 2, which covers the first two years of the time-horizon (i.e. the first one, with no change requests, and the second one, having a cash flow identical to all other years in the period).

	year	*1*	*2 and onwards*
Time savings interface development	PM	2.4	2.4
Time savings V&V	PM	0.6	0.6
V&V costs (Assessor) savings	Euro	3,000	3,000
Assumed PM time saving development change request	40%		
Time savings interface development (change request)	PM	4.8	4.8
Staff cost supplier	Euro/PM	6,000	6,000
Development and V&V cost savings (single interface)	Euro/year	21,000	21,000
Development and V&V cost savings (single change request)	Euro/year		35,400
Potential cost savings (5 interfaces)	*Euro/year*	105,000	105,000
Assumed cost savings (interface + change requests)	**40%**	**21,000**	**162,600**
	30%	21,000	133,800
	20%	21,000	105,000

Table 2: Calculation of savings in software development and V&V

The Financial Analysis performed on the case study demonstrated that, if cost savings enjoyed by suppliers are passed on to prices, the IM faces net cash flow savings over a multi-annual time horizon (assumed 15 years): comparing additional investment and operating costs with savings, the Net Present Value (NPV) of the adoption of FM is 50,917 Euro, with a 17.9% Internal Rate of Return [10]. Such values demonstrate the financial feasibility of the adoption of FM from the point of view of a single IM. The definition of NPV commonly used in CBA has been adopted. NPV, indicating the value of an investment as discounted to present time's values, is defined as:

$$NPV = \sum_t B_t(1+i_t)^{-t} - \sum_t C_t(1+i_t)^{-t} - K$$

where B indicate benefits, C costs, discounted with the rate i every year t of the project's lifetime, and K the initial investment.

Four scenarios were built considering the current operation on two Italian lines (a high-speed line and a highly congested node) to assess benefits for users in case cancellations or delays are avoided due to higher maintenance efficiency generated by FM. The simulation has assumed a service disruption causing 60' delay or train cancellation during one day on both reference Italian lines, for which a daily traffic of 116 trains (Milano-Melegnano) and 109 trains (Firenze-Bologna HS) is reported in the Network Statement [15].

The Economic Analysis has assessed the benefits due to higher maintenance efficiency, higher service availability, and time saved for a lower probability of service disruption. The assessment was based on the quantification of service disruptions that may happen on a rail line due to failure of RBC/RBC handover interface, and in particular those due to ambiguity of specifications. They are very rare according to 4SECURail Consortium's knowledge (0.1% of total cases). The calculation of penalties has been based on the Performance Regime in force on the Italian rail Network, issued by RFI (Italian IM) and valid until 2023 [15]. Values are the following:

- Delay 60 minutes: 4.5 € per minute = 300 € (applied both on HS lines and rail nodes).
- Train cancellations: 120 € (applied both on HS and regional services). The related amount saved by the IM is taken into account as net benefit in the CBA. Moreover, avoided service disruptions mean avoided delays for passengers, which can be monetized applying the appropriate Value of Time (VoT), defined in [6].

According to assumptions and calculation made, the service disruptions avoided have a value ranging from nearly 13,000 €/event in case of cancellations, 32-34,000 €/event in case the service disruptions cause 60' delays.

The annual value of time saved thanks to higher maintenance efficiency brought by the use of FM is estimated to range between 112,000 €/year in case of 60' delay on regional services, to 581,000 €/year in case of HS services cancellation. Those values rise to 562,000 €/year and 2.9 M€/year respectively,

[10] IRR is defined as the discounting rate necessary to obtain NPV=0. The indicator is adimensional and represents the expected return of the investment over the project's lifetime.

if change requests are taken into account. Figure 8 shows the order of magnitude of the annual (not discounted) value of each benefit category compared to costs. Not surprisingly, benefits from time saved for passengers are by far the most relevant benefit category (Figure 8). Indeed, expected benefits for users, although computed using many (realistic) assumptions, justify the adoption of FM and the necessary investment.

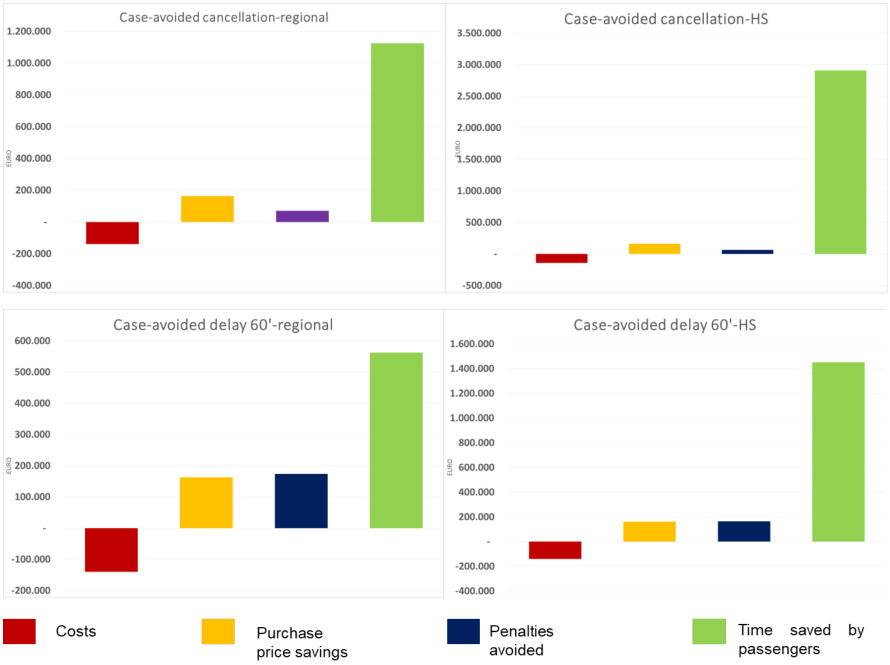


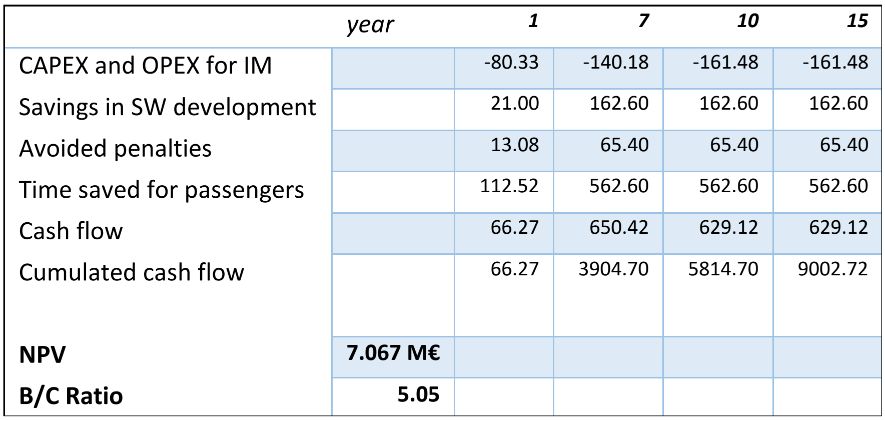
Fig. 8: Value of benefit categories per scenario (Euro/year)

The Economic analysis has demonstrated the net convenience of the FM adoption for the society as a whole (IM, users and all other involved stakeholders), since the net positive cash flow of benefits vs. costs during the 15-year period is about 9 MEuro. Indicators of the Economic analysis are highly positive: NPV is 7.067 MEuro and the Benefit/Cost Ratio (BCR) is 5.05, i.e., the process generates (actualized) benefits 5 times higher than cost borne by the IM. Such benefits are likely higher if FM are applied on a EU-27 scale. The net benefits for users and society may justify public granting of the adoption of FM in the railway safety domain. The definition of BCR is given as:

$$BCR = \frac{\sum_t B_t (1+i_t)^{-t}}{K + \sum_t C_t (1+i_t)^{-t}}$$

where variables have the same meaning as for the definition of NPV.

Cash flows of the Economic Analysis for relevant years are reported in the Table 3.



	year	***1***	***7***	***10***	***15***
CAPEX and OPEX for IM		-80.33	-140.18	-161.48	-161.48
Savings in SW development		21.00	162.60	162.60	162.60
Avoided penalties		13.08	65.40	65.40	65.40
Time saved for passengers		112.52	562.60	562.60	562.60
Cash flow		66.27	650.42	629.12	629.12
Cumulated cash flow		66.27	3904.70	5814.70	9002.72
NPV	**7.067 M€**				
B/C Ratio	**5.05**				

Table 3: Cash flow (kEuro/year) and Economic Analysis indicators

4 Discussion and Conclusions

The goal of the 4SECURail demonstrator was to show a possible way to use formal methods to improve the quality of System Requirement Specifications of signaling systems, using this experiment as a source of information on which to base a Cost-Benefit Analysis.

The outcomes of the project (see in particular [17]) have shown that the creation of an easy to understand and communicate, graphical but also executable, SysML model is an intermediate step that already allows to detect possible weaknesses in the natural language requirements, but that formal modeling and analysis is needed to detect and remove less trivial errors.

The results of the project have also shown how a ”formal methods diversity” approach can be successfully exploited to gain more confidence in the correctness of the formalization and analysis, and to gain access to a rich set of options for performing the analysis of the system. Even without using advanced formal verification techniques, e.g., involving bisimulations and complex temporal logic formulas, we have experienced that many easy-to-use analysis (e.g., static checks, invariants, deadlocks, reachabilities) can be be performed without any specific advanced background.

This lightweight use of formal methods is not aimed at full system verification/validation (which would not be possible anyway due to its parametric nature), but remains a very important aid for the early detection of ambiguities in the natural language requirements and in errors in their rigorous specification.

The Cost-Benefit Analysis developed on the 4SECURail demonstrator suggests that efforts and costs for formal analysis of system requirements are likely to be distributed among the various entities supporting the standard itself, and not to a single IM. Benefits are spread over the entire supply chain, including suppliers, if economies of scale are activated among IMs and suppliers in software development. The "multi-supplier" mode enabled by FM is likely to generate time and cost savings for rail safety industry.

The numeric cost/benefits results shown in the previous section are produced by the application of the Cost-Benefit Analysis procedure on data collected within the 4SECURail demonstrator test case, from the experience of the industrial partners, and from relevant literature (more details in [28]).

Although the compliance with the European Commission Guide to Cost-Benefit Analysis guidelines enhances methodological solidity, the input values used for the cost/benefit categories are derived from a limited basis of available data for such categories. This undermines the ability to generalize results to different case studies. Such threat to the *external validity* of the approach can only be addressed by enlarging the basis of available data by further experiments covering a wide spectrum of case studies, addressing different systems and employing different formal methods and tools. This would require greater attention of the formal methods community to the quantification of costs and benefits parameters (e.g., as given in [2]) since the evidence of the beneficial effects of formal methods is mostly given instead in the literature in a qualitative way. On the other hand, the analysis of the scarce literature where some cost quantification is given has shown how different the values of some cost categories can be in different formal methods frameworks (e.g., the software licenses when using commercial, qualified tools vs. using open-source ones). Nevertheless, we believe that the approach followed in the 4SECURail project can be taken as a first example of fully fledged Cost-Benefit Analysis, developed according to internationally accepted standards, on the application of Formal Methods in the rail signalling industry, that can be taken as an example on which to base further efforts in this direction.

Acknowledgements

This work has been partially funded by the 4SECURail project. The 4SECURail project received funding from the Shift2Rail Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No 881775 in the context of the open call S2R-OC-IP2-01-2019, part of the "Annual Work Plan and Budget 2019", of the programme H2020-S2RJU-2019. The content of this paper reflects only the authors'view and the Shift2Rail Joint Undertaking is not responsible for any use that may be made of the included information.

References

1. R. Aissat and A. Boralv. X2RAIL-2, Deliverable D5.3 Business Case, 2020.

2. D. Basile, A. Fantechi, and I. Rosadi. Formal analysis of the UNISIG safety application intermediate sub-layer - applying formal methods to railway standard interfaces. In *FMICS 2021*, volume 12863 of *LNCS*, pages 174–190. Springer, 2021.
3. D. Belli and F. Mazzanti. A Case Study in Formal Analysis of System Requirements. In *Software Engineering and Formal Methods. SEFM 2022 Collocated Workshops - LNCS 13765*, pages 164–173. Springer, 2022.
4. S Bibi, S. Mazhar, Minhas N.M., and I. Ahmed. Formal Methods for Commercial Applications Issues vs. Solutions. *Journal of Software Engineering and Applications*, 2014.
5. D. Burroughs. SNCF develops new-generation interlockings with a 1bn Argos partnership. https://www.railjournal.com/signalling/sncf-develops-new-generation-interlockings-with-e1bn-argos-partnership, 2018.
6. H. van Essen, L. van Wijngaarden, A. Schroten, D. Sutter, C. Bieler, S. Maffi, M. Brambilla, D. Fiorello, F. Fermi, R. Parolin, et al. Handbook on the external costs of transport, version 2019 1.1. *Delft: European Commission, Directorate-General for Mobility and Transport*, 2019.
7. EULYNX. Eulynx Project site, 2021. `https://eulynx.eu/`.
8. A. Ferrari and A. Fantechi et al. The Metro Rio case study. *Science of Computer Programming, Vol.78, Issue 7, 1 July 2013, Pages 828-842*, 2013.
9. A. Ferrari, F. Mazzanti, D. Basile, M. H. ter Beek, and A. Fantechi. Comparing formal tools for system design: a judgment study. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, pages 62–74, 2020.
10. John Fitzgerald, Juan Bicarregui, Peter Gorm Larsen, and Jim Woodcock. *Industrial Deployment of Formal Methods: Trends and Challenges*, pages 123–143. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
11. European Union Agency for Railways. Report on railway safety and interoperability in the EU. https://data.europa.eu/doi/10.2821/205360, 2018.
12. H. Garavel and M. H. ter Beek et al. The 2020 Expert Survey on Formal Methods. In *Formal Methods for Industrial Critical Systems. FMICS 2020, Proceedings, LNCS Vol 12327*, 2020.
13. M. Gleirscher and D. Marmsoler. Formal methods in dependable systems engineering: a survey of professionals from Europe and North America. *Empirical Software Engineering, Vol 25, Septmber 2020*, 2020.
14. A. Hall. Realising the benefits of formal methods. In *Formal Methods and Software Engineering, LNCS vol. 3785*, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
15. RFI Rete Ferroviaria Italiana. Prospetto Informativo della Rete, updated December 2021, with relevant annex "Gradi di Utilizzo dell'Infrastruttura: infrastruttura a capacità limitata e infrastruttura satura", February 2021.
16. J. Krasner. How Product Development Organizations can Achieve Long-Term Cost Savings Using Model-Based Systems Engineering (MBSE). https://docplayer.net/18566603-How-product-development-organizations-can-achieve-long-term-cost-savings-using-model-based-systems-engineering-mbse.html, 2015.
17. F. Mazzanti and D. Basile. 4SECURail Deliverable D2.2 "Formal development Demonstrator prototype, 1st Release ". https://www.4securail.eu/Documents.html, 2020.
18. F. Mazzanti, D. Basile, A. Fantechi, S. Gnesi, A. Ferrari, A. Piattino, L. Masullo, and D. Trentini. 4SECURail Deliverable D2.1 "Specification of formal development demonstrator". https://www.4securail.eu/Documents.html, 2020.

19. F. Mazzanti and D. Belli. 4SECURail Deliverable D2.5 ”Formal development demonstrator prototype, final release”. https://www.4securail.eu/Documents.html, July 2021.
20. F. Mazzanti and D. Belli. Formal Modeling and Initial Analysis of the 4SECURail Case Study. In *Proceedings of Fifth Workshop on Models for Formal Analysis of Real Systems, Munich, Germany, 2nd April 2022*, pages 118–144. Springer, 2022.
21. F. Mazzanti and D. Belli. The 4SECURail Formal Methods Demonstrator. In *Reliability, Safety, and Security of Railway Systems. Modelling, Analysis, Verification, and Certification: 4th International Conference, RSSRail 2022, Paris, France, June 1–2, 2022, Proceedings*, pages 149–165. Springer, 2022.
22. A. Piattino et al. 4SECURail Deliverable D2.3 ”Case study requirements and specification”. https://www.4securail.eu/pdf/4SR-WP2-D2.3-Case-study-requirements-and-specification-SIRTI-1.0.pdf, 2020.
23. A. Ruiz and B. Gallina et al. Architecture-driven, multi-concern and seamless assurance and certification of cyber-physical systems. In *Computer Safety, Reliability, and Security SAFECOMP 2016, LNCS vol 9923*, Cham, 2016. Springer International Publishing.
24. Shift2rail. 4SECURail (GA 881775) project site . http://www.4securail.eu.
25. UNISIG. FIS for the RBC/RBC Handover - SUBSET-039, year = 2015.
26. UNISIG. SUBSET-098 - RBC/RBC Safe Communication Interface, year = 2017.
27. UNISIG. Subset-037, euroradio fis v3.2.0, December 2015.
28. C. Vaghi. Table of CBA related bibliografy. https://zenodo.org/record/8174266.
29. C. Vaghi. 4SECURail Deliverable D2.4: ”Specification of Cost-Benefit Analysis and learning curves, Intermediate release”. https://www.4securail.eu/Documents.html, 2020.
30. C. Vaghi. 4SECURail Deliverable D2.6: ”Specification of Cost-Benefit Analysis and learning curves, Final release”. https://www.4securail.eu/Documents.html, 2021.
31. J. Woodcock, P.G. Larsen, J. Bicarregui, and J. Fitzgerald. Formal methods: Practice and experience. *ACM Computing Surveys, Volume 41 Issue 4, October 2009*, 2009.