

# Formal analysis and monitoring of legacy safety-critical interlocking systems with the use of certified industrial tools

Dalay Almeida<sup>1</sup>, Florian Jamain<sup>1</sup>, Thierry Lecomte<sup>1</sup>

CLEARSY, Aix-en-Provence - France  
dalay.almeida@CLEARSY.com, florian.jamain@CLEARSY.com,  
thierry.lecomte@CLEARSY.com

**Abstract.** Although Formal Methods have been used for decades in the development of industrial critical systems, there are still many products that do not use this technology. The use of Formal Methods in such a context is generally highly recommended, but not mandatory, as other technologies may be used as support to certify the safety of the systems. Relay-based railway interlocking systems, for instance, are legacy systems used in the majority of railway installations and whose safety has been attested through their use for decades. Their maintenance, however, requires analysis to avoid losing their safety features. In previous papers, we have presented the CLEARSY Safety Platform (CSSP) and how it can be used to analyse and replace these legacy interlocking systems in a safety-proved manner using certified industrial tools. In this paper, we extend this discussion to present how the CSSP can be used to monitor the legacy relay-based RIS to improve their safety during their execution. The strategy is to describe the system safety properties using logic and then implement it in the CSSP, which in turn is responsible for monitoring the system components to ensure its correct functioning and raise flags when an unsafe state is found. The benefits of using our approach in industry are discussed as we present how it can be applied in two industrial case studies.

**Keywords:** System Monitoring, Formal Specification, B-method, Relay-based Railway Interlocking Systems, CLEARSY Safety Platform

## 1 Introduction

Although the railway standards strongly recommend formal specification methodologies for the development of critical systems, there are still many products that do not use this technology. This may not always be a problem, as there are other manners to demonstrate the safety of a system. As an example, many legacy systems have been used for decades, which confer them a certain level of confidence, like in relay-based Railway Interlocking Systems (RIS). In previous papers [10, 15], we have discussed the difficulties of analysing these legacy systems and we proposed certified industrial tools (the CLEARSY Safety Platform - CSSP) and

a methodology to replace these RIS with computer-based systems created based on a formal development strategy using B-method. As a result, it is possible to generate safety-proved systems with the same behaviour as the legacy RIS.

Although the CLEARSY Safety Platform has presented some important results in the replacement of the relay-based RIS technology and logic with a smaller, cost-effective and safety-proved solution, it is important to consider that the industry still uses relay-based technology in the majority of RIS installations [24] as they are necessary in determined cases, notably in the connection between the computer-based systems and the railway equipment (train presence detection, signals, turnouts). So, replacing relay-based technology may not always be a solution. In such cases, the analysis of the safety of these systems remains a challenge as manual analysis cannot be trusted [13]. In this context, the industry needs a solution to analyse and monitor relay-based RIS to guarantee their safety without aiming at completely replacing them with new technology.

This paper presents an approach to apply CSSP in the analysis and monitoring of relay-based RIS. Based on a strategy for describing the behaviour of artificial intelligence [21], we propose describing the relay-based RIS expected behaviour in Propositional Logic. This behaviour can be extracted from the documentation of these systems, like the electrical circuit diagrams that represent their structure. As Propositional Logic is one of the bases of the B-method [1], the RIS logical description can be used in the formal specification necessary for the development strategy of the CSSP, as B is the main foundation of the CSSP system specification. Once this specification is proved and implemented, it is possible to connect the relay-based RIS components to the CSSP board inputs as a way to analyse if their states correspond to the expected behaviour. An approach to creating the monitor system using the certified tools and the benefits of using the CSSP monitor are discussed. Furthermore, two industrial cases are discussed and used as examples to present how important safety properties can be analysed and monitored with our approach. These case studies are based on the examples provided by the French National Railway Company (SNCF), discussed in previous works [10, 3].

Previously, we have proposed using propositional logic as the basis for specifying and implementing relay-based RIS as computer-based systems through a formal development strategy [10, 8]. Although this work has some similarities with these previous strategies, it proposes an alternative to those that do not want to replace the existing relay-based systems, allowing monitoring of legacy systems based on a formal strategy. Nonetheless, this work also does not invalidate our previous results on replacing legacy systems. The solution presented in this article is a complementary solution to be used in cases where the relay-based technology cannot be replaced and it does not diminish the importance of the previously proposed solutions.

The literature is scarce regarding the analysis of relay-based RIS when compared with computer-based strategies. Some works have proposed using formal methods to analyse these relay-based Railway Interlocking Systems [2, 5, 11, 12, 19, 22, 23, 25, 26], and we have analysed these works in previous publications [3,

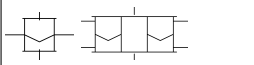
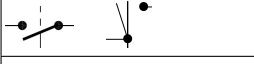
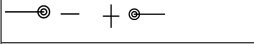
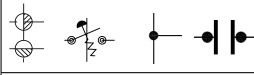
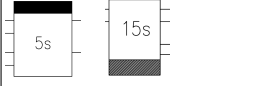
8–10]. The methodology we present in this paper stands out from the ones presented in the literature by proposing a different approach: instead of specifying the system to be implemented, our strategy is to analyse and constantly monitor existing systems with the use of certified tools based on a formal specification of their safety properties through a logical description of the system’s expected behaviour. By using our methodology, it is possible to provide formal means to analyse and monitor legacy relay-based systems based on Formal Methods and industrial tools with the highest level of safety certification (SIL4 [6]).

This paper is organised as follows. The sections 2 and 3 present some background of our work, providing some details about relay-based RIS and the CLEARSY Safety Platform, respectively. While Section 4 is devoted to presenting how the system safety properties may be logically modelled, Section 5 presents our strategy for modelling and implementing the CSSP monitor. Some case studies are discussed in Section 6, followed by a discussion about the benefits of industrial use of our approach in Section 7. A conclusion is provided in Section 8.

## 2 Relay-based RIS

Relay-based Railway Interlocking Systems are the implementation of interlocking systems using electrical circuits and components. A component is electrified (activated) when it is connected to the positive and negative source poles in a way that the electricity can flow through it. The most important component is the relay, which is composed of an electromagnet and one or more metallic contacts. When electrified, the relay produces an electromagnetic field that closes and opens contacts, controlling the electricity flow to other components. While monostables relays contain one coil that can pull or push contacts against gravity, bistable relays contain two coils that pull vertically-positioned contacts.

**Table 1.** Relay-based diagrams electrical components

	Monostable and bistable relays (coils), respectively.
	A monostable and a bistable contacts, respectively.
	Energy sources.
	A lever, a button, a junction and a capacitor, respectively.
	Blocks for timed activation and deactivation, respectively.

There are many different types of components. Some of them are depicted in Table 1. Each component has a different behaviour. As in this paper we focus on the activation/deactivation of components, we opted to focus on monostable relays, which is one of the most abundant components in the RIS electrical circuits. A more curious reader may be interested on reading our previous works in order to better understand the behaviour of other components [10, 8]. As our approach is based on the activation/deactivation of components, it can be applied to any type of electric component, thus, the function and behaviour of the components do not affect our methodology.

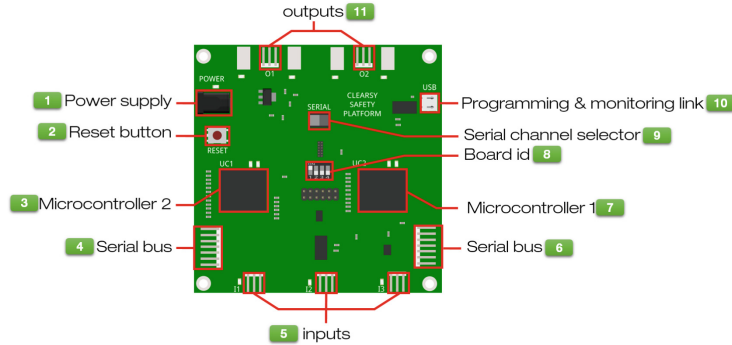
### 3 The CLEARSY Safety Platform (CSSP)

The B-method [1] has been industrially used for decades in the development of safety-critical systems [4]. It is strongly recommended by the industry standard for SIL4 software development [15]. By using the Atelier B [14, 7], a certified industrial tool to support the development, analysis and proof of systems, the industry has the necessary means to develop SIL3/SIL4 systems through a formal development strategy. However, regarding safety computers to run these applications, the market is limited: one may use PLCs (Programmable Logic Computers) or SIL3/SIL4-ready boards. Both of them bring a series of limitations that may hinder the development of systems [15].

In this context, the CLEARSY Safety Platform combines redundant hardware with proven software developed with B to create a solution for the development and implementation of SIL4 systems to be applied in critical environments. While producing our own hardware can significantly reduce costs, the use of Atelier B provides more freedom and control on software development compared to other solutions in the market [15]. The design of the CSSP was created to ease the certification process as each part of the hardware and development approach was designed to create and implement safety-critical systems.

The starting point of the development process is the B formal model which specifies the safety function to implement. This model is divided into an Abstract Machine, the abstract model based on logic, and an Implementation, the concrete refinement of the Abstract Machine. This implementable model is then translated by Atelier B using two different chains: a translation to C followed by a standard C compilation (gcc), and a translation to MIPS Assembly and then to the binary code. The software obtained is uploaded to the execution platform to be executed in parallel by two micro-controllers. The results obtained from both instances are compared to detect possible divergent behaviours. Besides, several other automatic safety analyses are provided within the IDE and the microcontrollers. The safety board design is presented in Figure 1. More information about the CLEARSY Safety Platform can be found at [15–18]. In this paper, we use the CSSP development kit containing: a physical version of the safety board and a CSSP version of Atelier B, which, in turn, contains the development interface (IDE), proof tools, translators and compilers, an interface

for the communication with the safety board and a safety board simulator that can be used when the physical version is not available.



**Fig. 1.** The CLEARSY Safety Platform Starter Kit 0 (SK0) - documentation available at <https://github.com/CLEARSY/CSSP-Programming-Handbook>

## 4 Logical description of the RIS behaviour

To use the CSSP as a tool to monitor Relay-based Railway Interlocking Systems, it is necessary to model the properties of these systems in the subset of the B language used by the tool. Besides, this formal model must be connected to the system to be analysed. In this section, we discuss an approach to model the relay-based Railway Interlocking Systems properties using propositional logic. We have already explored a similar strategy in previous work [10] to translate relay-based RIS to a software-based version of it. In this paper, however, this strategy is updated to provide a way to monitor relay-based RIS safety properties during the system execution instead of modelling the whole RIS. This section focuses on explaining how the relation between electrical components may be logically described so it can be implemented as a RIS monitor.

The approach discussed in this section is based on the logical description of system intelligence as discussed in [21]. The objective is to describe how the system must react to certain states by defining logical relations between the states of the components. In this context, one may use an implication ( $\Rightarrow$ ),

$$(A = \text{TRUE}) \Rightarrow (B = \text{TRUE})$$

to model that: if a component  $A$  is activated ( $\text{TRUE}$ ), a component  $B$  must also be activated. In this context, we use the Boolean values  $\text{TRUE}$  and  $\text{FALSE}$  to represent the activated and deactivated states of a component. This may be used, for instance, to state that if a train is detected in a certain portion of the tracks, the signals around it must be closed so no other train may enter the same portion of the tracks.

Similarly, one may use a bi-implication ( $\Leftrightarrow$ ) to represent RIS safety properties. In this case, the notation to be used is the following:

$$(A = \text{TRUE}) \Leftrightarrow (B = \text{TRUE})$$

In this context the state of both  $A$  and  $B$  are linked, meaning that one may never be updated without the other also being updated. This is important to represent components whose states follow the same rules, like signals that must always present the same information. Conversely, one may define that two components may never reach a specific state by using a negation (*not*) of a conjunction ( $\&$ ):

$$\text{not}((A = \text{TRUE}) \& (B = \text{TRUE}))$$

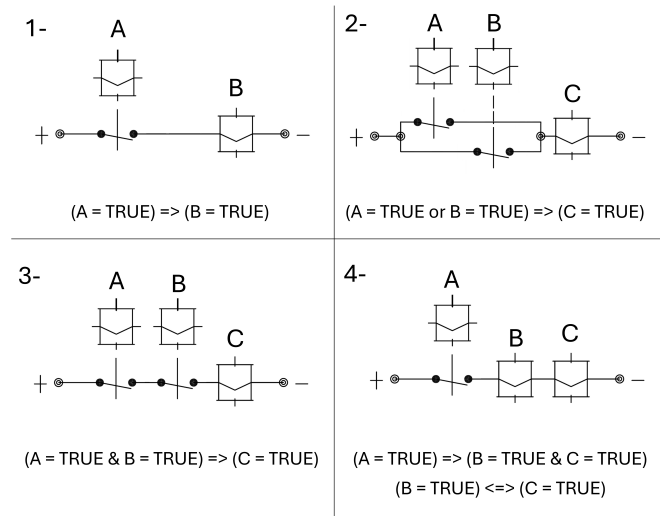
It is important to consider that more than two components may be involved in a safety property. For instance, two components may be responsible for the activation of a third one. In this context, conjunctions and disjunctions (*or*) may be used to add conditions. For instance, if a component  $C$  must be activated when the components  $A$  and  $B$  are activated, one may model it as:

$$(A = \text{TRUE} \& B = \text{TRUE}) \Rightarrow (C = \text{TRUE})$$

However, if only one between  $A$  and  $B$  must be true for the activation of  $C$ , one may model it as:

$$(A = \text{TRUE} \text{ or } B = \text{TRUE}) \Rightarrow (C = \text{TRUE})$$

Some examples of how the relation between component states can be logically described are presented in Figure 2.

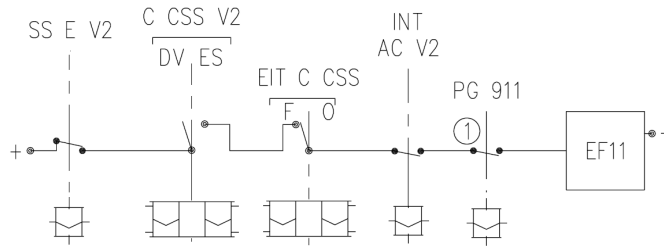


**Fig. 2.** Examples of logical description of some relay-based RIS configurations

Nonetheless, although formally modelling a complete relay diagram has been an interesting way to develop a safety-proved computer-based version of it, it does not respond to our need to monitor the existing relay-based ones. To do so, we may focus on the safety aspects of the system instead of modelling every component presented in a diagram. In this context, knowing the components and their function is essential to this task. For instance, in the example of Figure 3, the output *EF11* is responsible for permitting a signal related to a turnout to turn green. To send this information, the system must guarantee that no other train is using the turnout. The component *PG 911* is the pedal that detects train presence in this portion of the track and this component deactivation indicates that the track is occupied. Guaranteeing that the signal will not turn green if the track is occupied is a safety-related condition that may be written as:

$$(PG\_911 = FALSE) \Rightarrow (EF11 = FALSE)$$

This example is used in the remainder of this paper to illustrate the approach.



**Fig. 3.** Example of a component *EF11* (output) whose activation depends on the activation of a relay *PG 911*

The information about the function of the components and their system safety conditions is generally described in documents that are delivered with the system and/or during maintenance. So, we believe that this material can be used as a basis to describe the safety conditions in propositional logic. To use this logic in the B-method subset language used in the CSSP, one may apply the notation accepted by the tools. The next section describes how the RIS logical description can be specified in the CSSP tools to monitor relay-based RIS.

## 5 Creation and use of the RIS CSSP monitor

Once the safety properties of the system to be monitored are logically modelled, one may specify and refine them using the CSSP development kit. The states of the components considered in the logic are attached to the CSSP inputs. The core of this methodology is to physically connect the relay-based RIS components to the CSSP inputs. The CSSP may then constantly analyse the states of the components according to the logically modelled safety properties and then

output boolean information about the correctness of the system. This Section details all the steps regarding the development and use of the CSSP monitor.

The CSSP development approach divides the specification into an abstract machine and an implementation, respectively from the most abstract to the most concrete level of abstraction. The Atelier B has the tools to prove that the implementation is a correct refinement of the abstract machine. In our strategy for building a CSSP monitor, we focus on creating an abstract machine that contains the logic defined in propositional logic and an implementation that refines this logic. As we deal with the states of electrical inputs and outputs, instead of using the usual true and false boolean values, in our strategy we use `IO_ON` and `IO_OFF` to indicate the activation and deactivation of the inputs and outputs. In the CSSP model, these values belong to a basic type called `uint8_t`.

In order to formally specify the RIS monitor, one must focus on updating an output according to the modelled logic. In B, one may use the notation  $a : (l)$  to update a variable  $a$  according to a logical expression  $l$ . For instance, to specify the example mentioned in the previous section, one may write:

```
board_0_01 :( board_0_01 : uint8_t &
  ((board_0_I1 = IO_OFF) => board_0_I3 = IO_OFF) <=>
  (board_0_01 = IO_ON))
```

which updates the output variable `board_0_01` according to the CSSP established output type `uint8_t`, with the value `IO_ON` (activated) if the implication  $(\text{board\_0\_I1} = \text{IO\_OFF}) \Rightarrow \text{board\_0\_I3} = \text{IO\_ON}$  is true for the inputs `board_0_I1` and `board_0_I3`. These board inputs must then be physically connected to the components *PG 911* and *EF11*, respectively.

In this context, the strategy to specify the safety properties modelled in propositional logic is to follow the steps:

1. Update an output that indicates whether the monitored system is behaving correctly using the notation,  $a : (l)$
2. Type the output in the  $l$  part of the notation,
3. Condition the output value according to the defined logic using a bi-implication. This last step is also in the  $l$  part of the notation.

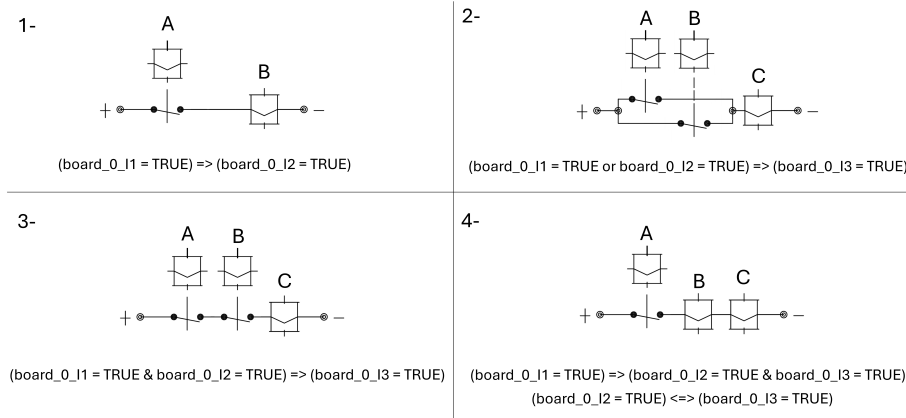
So, one may use the pattern

```
board_0_01 :( board_0_01 : uint8_t & (( <logic> ) <=>
  ( board_0_01 = IO_ON ))
```

to specify any safety condition in `<logic>` by replacing the variables of the propositional logic defined in the previous section by the board inputs and then output on `board_0_01` whether the safety property is met or not. Figure 2 may then be updated to present how the logic of some components configurations may be specified into the `<logic>` part of the pattern, as depicted in Figure 4. The renaming of the variables in the logic with the board inputs is important as the components are to be connected to these inputs at a later moment in our approach. The state of the output may also be updated during the system



specification: it is a developer's choice whether the output is activated or not when the safety property is not met. The purpose of the output value (activating other components or being read by a computer) plays an important part in this decision.



**Fig. 4.** Examples of CSSP specification of some relay-based RIS configurations, renaming the components with the board inputs, where A, B and C are attached to the board inputs `board_0_I1`, `board_0_I2` and `board_0_I3`, respectively

Once the abstract machine is specified with the safety properties to be monitored, one may implement it using the B-method strategy. When implementing this logic, the notation to be used is closer to those of programming languages. The Atelier B either automatically proves or provides the tools to prove that the implementation is coherent with the specification. A possible implementation of the logic used as an example throughout the paper initially creates variables that receive the states of the inputs and initialises the output of the board as `IO_OFF`:

```

VAR i1, i2 IN
  i1 :( i1 : uint8_t );
  i2 :( i2 : uint8_t );

  i1 <-- get_board_0_I1;
  i2 <-- get_board_0_I2;

  board_0_01 := IO_OFF;

```

Then, the logic can be implemented using conditions:

```

IF i1 = IO_OFF THEN
  IF i3 = IO_OFF THEN

```

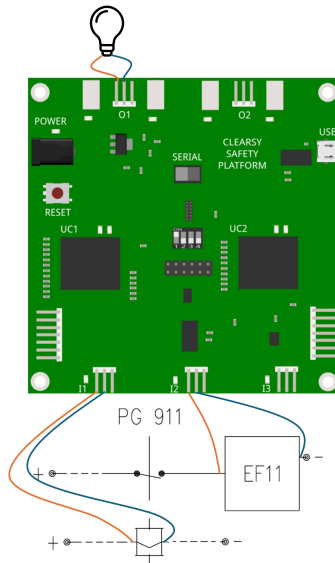
```

board_0_01 := IO_ON
END
ELSE
board_0_01 := IO_ON
END

```

The Atelier B is capable of automatically proving that this implementation is a valid refinement of the abstract machine.

Once the abstract machine and the implementation are defined, one may run the automatic compilation tool existing in Atelier B. By connecting the board to the computer, the tool can automatically translate and compile the specified system and upload it to the board, which may then execute it. Once the system is executing, one may connect it to the RIS electrical circuit to receive the states of the components and analyse them according to the specified logic (see Figure 5). As the RIS runs, the electrification of the wires activates the board inputs that are read by the implemented system. The monitor analyses if these inputs follow the modelled logic and then outputs the result of it. This output may then be read by a computer or it may trigger a light that may be used in the tracks to guide the train drivers. The use of lights to indicate signalling problems is already a normal practice in railway signalling. Some light panels contain this type of “system error” light in a pattern that is already known by the drivers [20].



**Fig. 5.** Example of connection between the RIS and the CSSP board

There are many different ways of implementing each propositional logic pattern we have presented as examples in this paper. As the implementation of the

specification is closer to programming and as the consistency of the implementation regarding the specification is completely proven, we are not presenting how each pattern may be implemented. The choice of how to implement the conditions to comply with the modelled logic is a matter of the developer’s choices. Nonetheless, in the next section, we discuss some examples and provide their specification elsewhere, which may be satisfactory to the more curious readers.

## 6 Case studies

In this paper, we apply our CSSP monitor strategy in two previously studied industrial case studies: the Temporary Reversed Direction Installation (ITCS - Installations Temporaires de Contre Sens) [10] and the Signalling Light Panel [3] examples, both used by the French National Railway Company. They exemplify how the CSSP monitor can be used to analyse the safety of the system according to logically specified safety properties. The case studies specification, implementation and the CSSP academic tools are available elsewhere<sup>1</sup>.

### 6.1 Temporary Reversed Direction Installation

The ITCS is a system that controls the signals related to a turnout. The railway tracks are divided into portions called Control Areas. In a two-way portion of the tracks, when one of the tracks is blocked, the trains must pass through the opposite-way tracks (see Figure 6), which may cause a collision. The signals must guarantee that only one train may access this "dangerous zone" at a time. The ITCS is responsible for the communication between the two involved control areas, informing whether the signals may go green according to the train’s presence and safety conditions. In this case study, we model and analyse the system in Control Area A. While the component *KIT C 911* is responsible for controlling the signal on the left, the component *EF11* is responsible for informing Control Area C whether its signal may turn green or not. The activation of each of these components generates permission to enter the dangerous zone, so the system may guarantee that these components are never activated at the same time.

By analysing the relay diagram and the documentation about it (not provided in this paper due to confidentiality reasons), there are many safety properties we can model. The most obvious is that the components *KIT C 911* and *EF11* must never be activated at the same time:

```
not((KIT_C_911 = TRUE) & (EF11 = TRUE))
```

Another safety property modelled in the relay diagram is the one used as an example throughout this paper. Nonetheless, we may extend the safety condition to consider both light signals. When the pedal *PG 911* detects a train in the dangerous area, the signals around it must be closed so no train has permission to enter. Thus, we can model this condition as:

```
(PG_911 = FALSE) => (KIT_C_911 = FALSE & EF11 = FALSE)
```

<sup>1</sup> <https://zenodo.org/records/11094051>

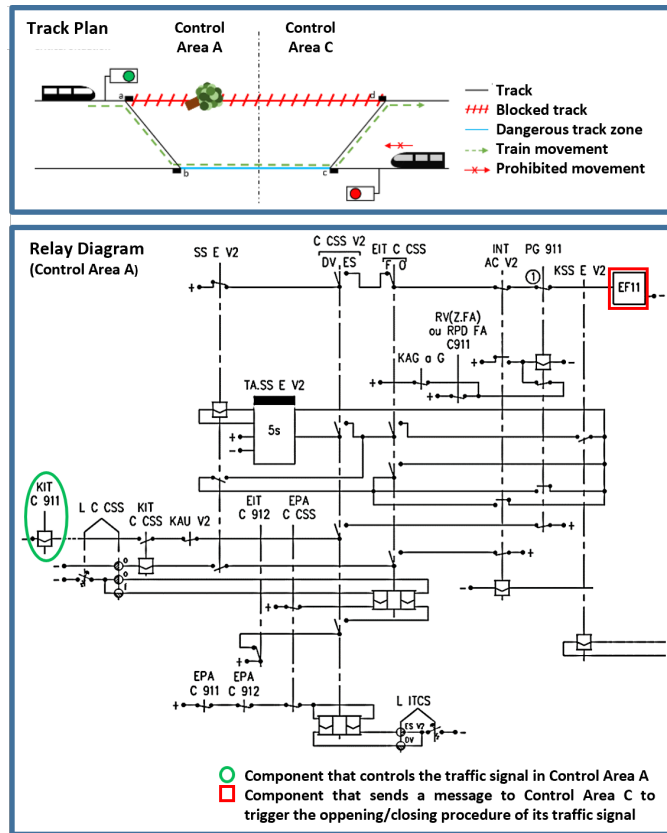
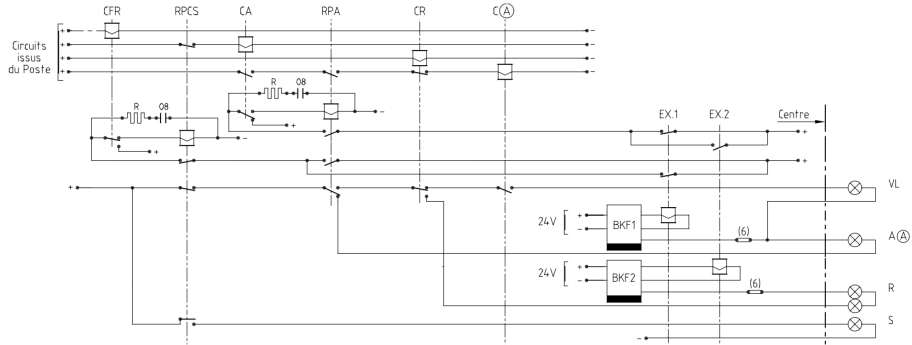


Fig. 6. The track plan and the relay diagram of the ITCS example

## 6.2 Signalling Light Panel

The control of a signal panel must meet many safety properties. Figure 7 depicts one example of a relay-based schema of such a control system. There are many different types of signal panels with different types of lights. In our case study, our panel has five lights that are used to communicate four types of information. The lights *VL*, *A* and *S* are, respectively, the green, yellow and red lights. The functions of these lights follow the pattern: green for movement authorisation, yellow for attention and red to stop. A pair of yellow lights is represented by *R* in the diagram and it has the function of informing the train driver to limit its velocity to 30km/h.

When analysing the control of signal panels, it is important to understand that when a problem is found, the system must seek a safer state. In a light panel, the safest state is always the red light, i.e., when the train has no permission to move. In this context, the relay *RPCS* has the task of turning off all other



**Fig. 7.** Relay diagram of the signal panel case study

lights and turning only the red one on when a problem is found. If this relay is deactivated, the red light must be on:

$$(RPCS = FALSE) \Rightarrow (S = TRUE)$$

Similarly, the relay *RPA* is responsible for activating the yellow light when the green light is not available as the former is already safer than the latter. This scenario occurs when the system does not detect a problem dangerous enough to trigger the red light (*RPCS* is on), but still it requires some attention from the driver. In this context, we may define the following safety property:

$$(RPCS = TRUE \ \& \ RPA = FALSE) \Rightarrow (A = TRUE)$$

A last important safety property is focused directly on the lights: if the green and all yellow lights are off, the red light must be on. This verification is redundant as the analysis involving the relays *RPCS* and *RPA* already analyses the state of the red light when the other cannot be turned on by the relays. Nonetheless, monitoring the system regarding this safety property is still important as a dysfunctional analysis to avoid false positives due to component failures or short circuits. This safety property can be monitored by connecting the CSSP board to all lights and make the following verification:

$$(VL = FALSE \ \& \ A = FALSE \ \& \ R = FALSE) \Rightarrow (S = TRUE)$$

This is an example that can be easily specified, implemented and executed with the industrial version of the CSSP board. However, as the academic version of the CSSP (available for research purposes) contains only three inputs, this analysis requires a new approach: the conjunction between the states of *VL*, *A* and *R* is calculated in one CSSP board *CSSP-1* that sends the resulting boolean value to another CSSP board *CSSP-2* (connection between the output of *CSSP-1* and the input of *CSSP-2*) that may then receive the state of *S* and calculate the implication. As the strategy discussed in this paper is focused on industrial purposes, the application of the monitoring of this safety property does not require such a specific approach. Thus, this example is not available to

the reader of this paper. Nonetheless, it serves as a reminder that the industrial CSSP board has twenty-four inputs or outputs, which is more than enough to be used to monitor such industrial examples. Thus, one single CSSP board can be responsible for analysing several safety properties at the same time by addressing the correct inputs and outputs when specifying the safety properties logic.

## 7 Discussion

The case studies presented in this paper exemplify many safety properties that can be analysed and monitored using our CSSP strategy. Once the CSSP board is attached to the system to be monitored, it is possible to constantly analyse the system execution based on the components states. When the monitor detects that a safety property is not met, it raises a flag that may trigger a signal light to the driver and/or that may be read by computers connected to a control centre responsible for triggering a repair team. In this section, we present how detected problems in different safety properties may be interpreted and how the CSSP monitor can be applied to benefit the industry.

Our initial concern is to formally analyse existing systems without aiming at replacing them with computer-based systems. The documentation of these systems is generally focused on the safety properties that they must meet, so an analysis of these properties is the most essential task in our process. By using the CSSP, it is possible to support the in-loco tests that are performed during the maintenance of the legacy relay-based RIS. In this context, it is possible to provide a formal analysis of the system's safety properties by using the certified safety board and development approach to create a monitoring application. In this context, although we are dealing with a legacy system that was not conceived with Formal Methods, one is still capable of ensuring it follows the defined safety properties using Formal Methods.

Another way of using the CSSP Monitor follows the same principle, however, instead of analysing the system during in-loco tests, one may use the CSSP to constantly monitor the system regarding these safety properties. In this context, it is possible to obtain a constant analysis of the system regarding the defined safety properties. If the monitor is connected to a "defect" light, it can signal to the train driver that the system is defective when the safety conditions are not met, and then the driver may take standard precautions. When the monitor is connected to a computer, it may be used to inform the concerned engineers that the system needs maintenance as it is not behaving properly. For instance, if, for some reason, the outputs *KIT C 911* and *EF11* of the ETCS example are activated at the same time, it may turn on the "defect" light of the signals. As a result, the driver may be immediately informed that the information presented in this signal may not be trusted. Then, as the CSSP is connected to a computer that sends the output information to a control centre, the engineers may send people to analyse the system and find the defects. In this context, the CSSP monitor may not only avoid a dangerous situation but also provide the necessary information for engineers to take action and repair the system.

Nonetheless, it is important to observe that safety problems may not only be created during the conception or maintenance phases. The engineering behind relay-based RIS is constantly concerned about dysfunctional problems that may affect the system. This is the main reason why the components must be highly certified to endure stress and time. As a way to create a second layer of protection, the CSSP analysis may also be used to provide a constant dysfunctional analysis of the system, raising a flag when components with linked behaviours are not working properly. As an example, the pedal *PG 911* is responsible for controlling the state of the output *EF11* through one of its contacts. In a case where the mechanical parts of the relay fail, the CSSP is capable of detecting the inconsistency between the states of the relay and the component *EF11* according to the defined safety property.

Although we use real industrial examples as case studies, the approach presented in this paper has not yet been used in an industrial context. Monitoring systems based on our formal approach is still an idea that may become a product shortly. The proposed specification of the safety properties is based on the documentation of the system, which may vary according to the company. Some examples of documentation are: the relay diagrams, the system requirements specification and the Domain-Specific Languages (DSLs) used to model the systems. To specify the safety properties of our case studies, we used as a basis the relay diagrams and other documents that detail the components and their function in the system. Other case studies from other contexts may need different approaches. Nonetheless, in such a railway context, a document that specifies the system requirements is commonly available. The specification of the system properties in propositional logic, their translation into B and the refinement of the specification must be made manually, as it depends on the knowledge about the system safety. The automation of the approach relies on the automatic proof, implementation and execution of the monitor using the CSSP board.

## 8 Conclusion

This paper presents an approach to using the CLEARSY Safety Platform to monitor existing relay-based Railway Interlocking Systems. Differently from our previous works, instead of proposing the transformation of such legacy systems into computer-based ones, we propose the use of the certified CSSP tools and development methodology to formally analyse and monitor the RIS regarding safety properties. Our solution may be used to improve the quality of legacy manually analysed relay-based systems by allowing their formal analysis and monitoring based on Formal Methods. Besides, the monitor may be used to provide constant safety and dysfunctional analysis to guide the drivers and trigger repair teams with minimal delays. Two industrial case studies are presented to exemplify how our approach may be used in an industrial context.

In the future, we aim to study how the CSSP may be applied in other contexts to analyse and/or rebuild safety-related systems. We are now aiming at applying it to analyse systems related to the safety aspects of electricity production.

Besides, we aim to improve our methodology to be able to analyse and monitor safety-critical computer-based systems as well. As an industrial product, we are in discussion with many industrial and academic partners from other areas to discover new applications of such a flexible and useful tool in systems different from those where we are applying them right now. Future publications may result from these partnerships.

## References

1. Jean-Raymond Abrial, Matthew Lee, David Neilson, P. Scharbach, and I Sørensen. The b-method. In *International symposium of VDM Europe*, pages 398–405. Springer, 1991.
2. Arturo Amendola, Anna Becchi, and et. al. Cavada. Norma: a tool for the analysis of relay-based railway interlocking systems. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 125–142. Springer, 2022.
3. PER Bezerra, MVM Oliveira, Thierry Lecomte, and DI de Almeida Pereira. Csp specification and verification of a relay-based railway interlocking system. In *Brazilian Symposium on Formal Methods*, pages 36–54. Springer, 2023.
4. Michael Butler, Philipp Körner, Sebastian Krings, Thierry Lecomte, Michael Leuschel, Luis-Fernando Mejia, and Laurent Voisin. The first twenty-five years of industrial use of the b-method. In *International Conference on Formal Methods for Industrial Critical Systems*, pages 189–209. Springer, 2020.
5. Roberto Cavada, Alessandro Cimatti, Sergio Mover, Mirko Sessa, Giuseppe Cadavero, and Giuseppe Scaglione. Analysis of relay interlocking systems via smt-based model checking of switched multi-domain kirchhoff networks. In *2018 Formal Methods in Computer Aided Design (FMCAD)*, pages 1–9. IEEE, 2018.
6. EN Cenelec. 50128-railway applications-communication, signalling and processing systems-software for railway control and protection systems. *Book EN*, 50128, 2012.
7. ClearSy. *Atelier B User Manual, version 4.0*. ClearSy System Engineering, Parc de la Duranne - 320 av. Archimède - Les Pléiades III Bat A - 13857 AIX EN PROVENCE CEDEX 3 - FRANCE.
8. Dalay Israel de Almeida Pereira. *Analysis and formal specification of relay-based railway interlocking systems*. PhD thesis, Centrale Lille Institut, 2020.
9. Dalay Israel de Almeida Pereira, Sana Debbech, Matthieu Perin, Philippe Bon, and Simon Collart-Dutilleul. Formal specification of environmental aspects of a railway interlocking system based on a conceptual model. In *International Conference on Conceptual Modeling*, pages 338–351. Springer, 2019.
10. Dalay Israel de Almeida Pereira, David Deharbe, Matthieu Perin, and Philippe Bon. B-specification of relay-based railway interlocking systems based on the propositional logic of the system state evolution. In *International Conference on Reliability, Safety, and Security of Railway Systems*, pages 242–258. Springer, 2019.
11. Shiladitya Ghosh, Arindam Das, Nirvik Basak, Pallab Dasgupta, and Alok Katiyar. Formal methods for validation and test point prioritization in railway signaling logic. *IEEE Transactions on Intelligent Transportation Systems*, 18(3):678–689, 2016.
12. Anne E Haxthausen, Andreas A Kjær, and Marie Le Bliguet. Formal development of a tool for automated modelling and verification of relay interlocking systems.



- In *FM 2011: Formal Methods: 17th International Symposium on Formal Methods, Limerick, Ireland, June 20-24, 2011. Proceedings 17*, pages 118–132. Springer, 2011.
13. Anne E Haxthausen, Marie Le Bliguet, and Andreas A Kjær. Modelling and verification of relay interlocking systems. In *Monterey Workshop*, pages 141–153. Springer, 2008.
  14. Thierry Lecomte. Atelier b. *Formal Methods Applied to Complex Systems: Implementation of the B Method*, pages 35–46, 2014.
  15. Thierry Lecomte. Programming the clearsy safety platform with b. In *Rigorous State-Based Methods: 7th International Conference, ABZ 2020, Ulm, Germany, May 27–29, 2020, Proceedings 7*, pages 124–138. Springer, 2020.
  16. Thierry Lecomte, David Deharbe, Paulin Fournier, and Marcel Oliveira. The clearsy safety platform: 5 years of research, development and deployment. *Science of Computer Programming*, 199:102524, 2020.
  17. Thierry Lecomte, David Déharbe, Denis Sabatier, Etienne Prun, Patrick Péronne, Emmanuel Chailloux, Steven Varoumas, Adilla Susungi, and Sylvain Conchon. Low cost high integrity platform. *arXiv preprint arXiv:2005.07191*, 2020.
  18. Thierry Lecomte, Bruno Lavaud, Denis Sabatier, and Lilian Burdy. A safety flasher developed with the clearsy safety platform. In *Formal Methods for Industrial Critical Systems: 25th International Conference, FMICS 2020, Vienna, Austria, September 2–3, 2020, Proceedings 25*, pages 210–227. Springer, 2020.
  19. Ahmad Mirabadi and Mohammad Yazdi. Automatic generation and verification of railway interlocking control tables using fsm and nusmv. *Transport Problems*, 4:103–110, 2009.
  20. Roger Rétiveau. *La signalisation ferroviaire*. Presse de l’école nationale des Ponts et Chaussées, 1987.
  21. Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3 edition, 2010.
  22. Xiaoli She, Yun Sha, Qiang Chen, and Jian Yang. The application of graphic theory on railway yard interlocking control system. In *2007 IEEE Intelligent Vehicles Symposium*, pages 883–887. IEEE, 2007.
  23. Pengfei Sun, Simon Collart-Dutilleul, and Philippe Bon. A model pattern of railway interlocking system by petri nets. In *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 442–449. IEEE, 2015.
  24. Gregor Theeg and Sergej Vlasenko. *Railway signalling & interlocking: International compendium*. 2019.
  25. PHJ Van Eijk. Verifying relay circuits using state machines. *Logic Group Preprint Series*, 173, 1997.
  26. Kirsten Winter. Model checking railway interlocking systems. *Australian Computer Science Communications*, 24(1):303–310, 2002.