COBEM 2023 27TH INTERNATIONAL CONGRESS OF MECHANICAL ENGENEERING

ABCM Brazilian Society of Mechanical Sciences and Engineering
27th ABCM International Congress of Mechanical Engineering
December 04-08, 2023, Florianópolis-SC, Brazil

COB-2023-0915
MODEL CHECKING AIDED DESIGN OF ALARM AND SEAL LOGIC FOR SHIP CONTROL AND MONITORING SYSTEM

Rafael Celestino dos Santos
Universidade Federal de Santa Catarina, CTC-DAS, Florianópolis - SC, Brasil, 88040-900.
Marinha do Brasil, Rio de Janeiro - RJ, Brasil, 21931-095.
celestino@marinha.mil.br

Max Hering de Queiroz
Universidade Federal de Santa Catarina, CTC-DAS, Florianópolis - SC, Brasil, 88040-900.
max.queiroz@ufsc.br

Abstract. *This paper presents an application of model checking (MC) to the design of Control and Monitoring System (CMS) of the Brazilian Navy (Marinha do Brasil - MB) Ships, with the objective of validating the safety of the specification of the alarm and seal logic implemented in Programmable Logic Controllers (PLC). The CMS is responsible for monitoring and controlling the propulsion plant and mitigating possible damage to the ship, such as fire and flooding. The alarm and seal logic aims to retain the alarm signal in the PLC until it is remotely recognized by the operator in the Human Machine Interface (HMI). The use of MC is recommended by the IEC 61508 and IEC 61511 standards for critical programmable safety systems, such as the CMS, because it is an automatic and exhaustive formal verification technique that allows validating whether the safety properties are fulfilled by the system, for all possible scenarios. A functional error in the CMS can cause a risk to the lives of the crew, material damage to the ship and environmental risks. The specification of the control logics in the MB is elaborated through Binary Logic Diagram (BLD), according to ISA 5.2. The MC aided design methodology follows these steps: 1) the safety requirements are represented by Cause and Effect Matrix (CEM), according to IEC 62881; 2) each CEM safety property is translated into two Linear Temporal Logic (LTL) formulas representing Spurious Trip Freeness (STF) and Failure on Demand Freeness (FDF), which are established by IEC 61511-1; 3) the BLD model is translated into formal Temporized Transition System (TTS), by means of the intermediary language FIACRE; 4) the LTL properties are verified in the TTS model by means of the model checker TINA/SELT. The results show that the BLD for the Alarm and Seal logic proposed by the MB design team, fulfills the FDF property, but not the STF property. Additionally, TINA/SELT presents a counterexample, from which it is possible to identify the error and to correct the BLD. After correction, a new verification has been performed. The new result shows that the corrected BLD fulfills the STF and FDF properties. In this way, the MC proves to be an efficient technique to aid engineers in identifying errors even in the early stages of the CMS design, reducing the correction costs and increasing reliability.*

Keywords: *: Model Checking, Programmable Logic Controller, Binary Logic Diagram, Binary Logic Diagram, Cause and Effect Matrix, Linear Temporal Logic.*

1. INTRODUCTION

The Brazilian Navy continuously seeks to update its naval resources (ships, submarines, aircraft and equipment for the Marines) with modern and reliable systems and equipment, by obtaining scientific and technological knowledge. Among the projects developed by MB, the CMS stands out for being responsible for monitoring and controlling the propulsion plant and possible damage to ships, such as fire and flooding. Regarding complexity, the system has two Programmable Logic Controllers to control three subsystems: the Propulsion and Auxiliary Control and Monitoring Subsystem (SCMPA), the Damage Control Subsystem (SCAV) and the Remote Manual Subsystem (SMR). As for the criticality, the CMS is considered a critical system because a functional error in the CMS can cause a risk to the lives of the crew, material damage to the ship and environmental risks.

The development of the CMS is a challenging task, especially regarding the elaboration of the specifications and the programming of the PLC control logics to ensure that the safety requirements are fulfilled by the system. Currently, the CMS design is verified and validated through tests and simulations. These validation techniques are important and necessary, but they are not exhaustive, that is, they do not reach all possible states of the system, only a limited number of scenarios are verified.

Due to this limitation, several technical standards have emerged to support the development, verification and validation of programmable safety systems (Gall, 2008), such as IEC-61508 (2010) e IEC-61511 (2023). These standards recom-

mend the use of formal verification for the safety of electrical, electronic and programmable systems considered critical, as a complement to tests and simulations, in order to reduce to acceptable levels the possibility of failures in the system.

The formal verification is the mathematical demonstration, through mathematical methods, used to prove whether a model of the system at an appropriate scale satisfies certain properties or not (Wing, 1990). Model Checking is an automatic and exhaustive method of formal verification that mathematically explores the entire state space of a system. It aims to prove whether a specific property holds true in all possible scenarios. If the desired property is not satisfied, a counterexample is provided (Clarke, 1997).

In the literature, there are many studies on the application of MC to PLC-controlled systems, aiming to detect errors in various stages of the design process. Most of the works apply the MC in the PLC programming stage: Farines *et al.* (2011) present an automatic verification chain formal for the programs (written in the Ladder language) of an automatic pneumatic system; Adiego *et al.* (2015) propose a formal verification methodology, using MC, with the objective of finding errors in PLC programs applied to critical systems of the nuclear industries; Chadwick *et al.* (2018) present MC-based automatic verification modeling for application in train control systems, in order to verify that safety interlocking requirements are fulfilled by the system; Reis *et al.* (2018) present an automated method for formal verification of PLC programs integrated to the development methodology of Safety Instrumented Systems (SIS) in oil and gas industry.

Other works focus on the application of MC in the domain of conceptual design (specification of control logic), a step before programming. According to Liggesmeyer *et al.* (1998), quoted by (Gergely *et al.*, 2011), in PLC-controlled systems, correcting errors at this stage of the project is more financially advantageous than at other stages. The following papers present this characteristic: Pakonen and Björkman (2017) propose a MC-based methodology for applications in Finnish nuclear industries, aiming to identify errors due to spurious activation; and Lázaro *et al.* (2019) propose a methodology for MC aided development of CMS for the ships of the Brazilian Navy. This methodology focuses on the application of MC in the conceptual design stage and was validated through a real case study: design of the emergency stop logic of a ship's main engine.

The objective of this article is to extend the validation of the methodology proposed by Lázaro *et al.* (2019), through another real application of the CMS: design of alarm and seal logic. The structure of the paper is organized as follows. Section 2 shows an overview of MC aided design of ship control and monitoring sytem. Section 3 presents the alarm and seal logic and the application of the MC-based methodology in the design of logic for CMS. Section 4 presents the results and analysis. Finally, the article ends with the conclusion about the new application of the methodology.

2. MODEL CHECKING AIDED DESIGN OF SHIP CONTROL AND MONITORING SYSTEM

2.1 Control and monitoring system

The Control and Monitoring System (CMS) is the system responsible for commanding and monitoring the main machines and accessory systems of the ship propulsion and damage control system. The CMS is composed of 3 sub-systems: 1) Propulsion and Auxiliary Control and Monitoring Subsystem (SCMPA) that aims to monitor and control the ship propulsion, providing setpoints for the engine and turbine regulators. This subsystem also monitors and acts on the auxiliary equipment of the ship, such as: pumps and valves; 2) Damage Control Subsystem (SCAV) is responsible for monitoring high temperature, occurrence of flooding and the presence of smoke in the various compartments of the ship. When an abnormal situation is detected, the subsystem alarms and acts on some equipment remotely, such as the CO2 fire suppression system and fire sprinkler system; and 3) Remote Manual Subsystem (SMR) is a backup system for operation in degraded mode, with the minimum conditions necessary for the operation of the ship.

The CMS operates with 2 redundant PLCs in hot standby, in order to increase the reliability of the system. The main PLC actively operates on the process variables to keep them within established limits, while the secondary PLC operates passively by reading the input variables of the system and only acting when the main PLC fails. The supervisory system (Supervisory Control and Data Acquisition - SCADA) is composed of independent computers (consoles), distributed throughout the operation centers of the ship, whose purpose is to allow CMS operators to monitor process variables and intervene when necessary. When the variables of the CMS exceed the defined range of values, the PLC identifies abnormal situations and generates an alarm for each system variable. The supervisory system provides the alarm information on the console screen for the operators. Given the information, operators can recognize the alarms and act on the system, aiming to restore the situation to normality.

2.2 Actual methodology for development of CMS

The Figure 1 shows the actual methodology for developing CMS projects used at MB. The inputs for specifying the PLC code are the operating specifications and the experience of the design team. The operational specification contains information in natural language, flowcharts of the desired functionalities and operational and safety requirements of the system. With this information, designers specify the PLC code through Binary Logic Diagram (BLD), in accordance with the ISA-5.2 (1992). After this, the programming team interprets the BLD and generates the code for the PLC, using one of

the programming languages specified in IEC-61131.3 (2003). With the PLC code prepared, the technical team performs the tests in order to validate the system. If the system has no errors during testing, the CMS is validated. In case of errors, the PLC code is rejected and revised by the project team, with subsequent performance of new tests.

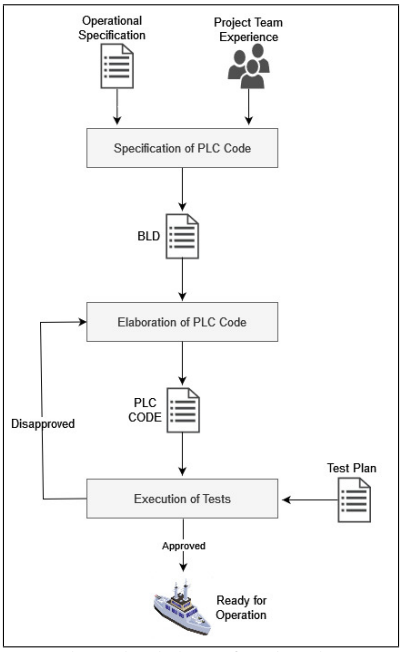

Figure 1. Actual methodology for development of CMS.

2.3 Overview of the methodology proposed by Lázaro *et al.* (2019)

The Figure 2 shows an overview of the formal verification methodology proposed by Lázaro *et al.* (2019) to aid the CMS development project. This methodology differs from the methodology currently used by MB due to: 1) the inclusion of a formal verification chain using MC; and 2) the use of Cause and Effect MatriX (CEM) to represent the safety requirements of the CMS, aiming to facilitate the understanding of the system by members of the design and operation teams.

The part highlighted in Fig. 2 represents the formal verification chain. This chain integrates with the actual methodology of the CMS, in order to complement the tests and simulations in the validation of the system. The verification chain receives inputs from the BLD and CEM, which express the system behavior and safety requirements of the CMS. The BLD are translated into a FIACRE intermediate language (Berthomieu *et al.*, 2008) that preserves the temporal and behavioral characteristics of the system. The FIACRE language is automatically translated into the low-level verification formalism called TTS (Timed Transition System), through the FRAC tool. The requirements defined in the CEM are translated into formulas of Linear Temporal Logic (LTL). The TINA/SELT model checker (computational formal verification tool) receives the TTS and LTL formalisms (Berthomieu and Vernadat, 2006). Finally, TINA/SELT automatically and exhaustively checks that all safety properties are satisfied by the system. If result shows that the properties are satisfied, PLC code is elaborated. Otherwise, the TINA/SELT presents a counterexample, from which the technical team of MB can correct the BLD.

2.4 Construction of Cause and Effect Matrices

Cause and Effect Matrix (CEM) is a matrix-based form of representing the logic of dependencies between causes (events) and effects (actions) that should which must occur automatically in the system. The IEC-62881 (2018) addresses the configuration and implementation of CEM for consistent use in engineering activities. The main advantage of this formalism is its simplicity in representing the requirements that the system must fulfill, facilitating the exchange of information between members of design and maintenance teams.

The Figure 3 shows the structure of the CEM, which is a table where the rows of the matrix correspond to the inputs (causes) and the columns correspond to the outputs (effects). The intersections are typically represented by symbols that express boolean logic operations (such as "AND", "OR", "NOR") and timed logic. However, the IEC 62881 standard does not establish a standard set of symbols. Therefore, each sector can define the representation of logical operators that best suits their project. Lázaro *et al.* (2019) proposed the following symbology for CEM, which express the safety requirements of the CMS development project:

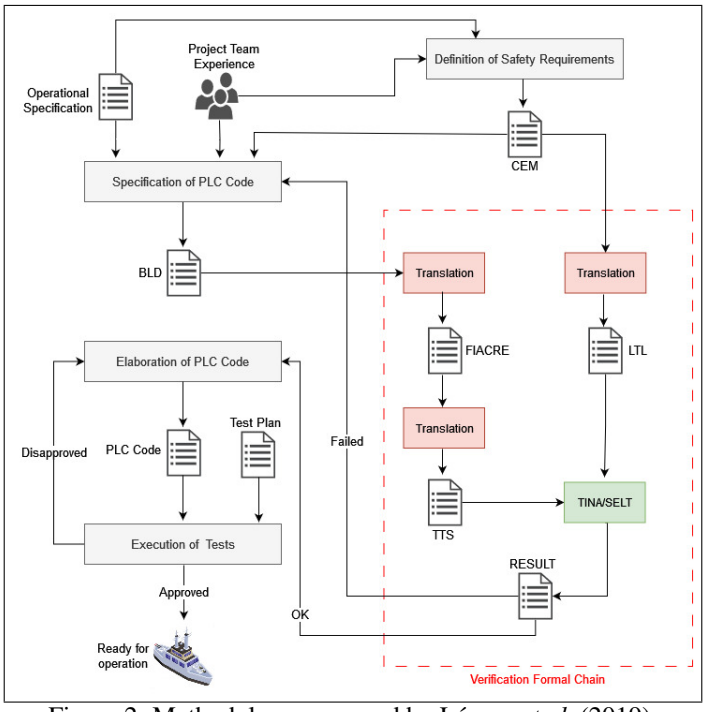

Figure 2. Methodology proposed by Lázaro *et al.* (2019).

a) X: represents the logical OR between the related causes;
b) N: represents the OR NOT logic between the related causes;
c) A*i*: represents the AND logic between causes related to index $i = 1, 2, ...$;
d) NA*i*: representa a lógica AND NOT entre as causas relacionadas ao índice $i = 1, 2, ...$;
e) T*x*: the causes related by this symbol should remain active for x seconds to activate the effect;
f) +: Representation of the logics that activate the retentive effect;
g) -: Representation of the logics that deactivate the retentive effect.

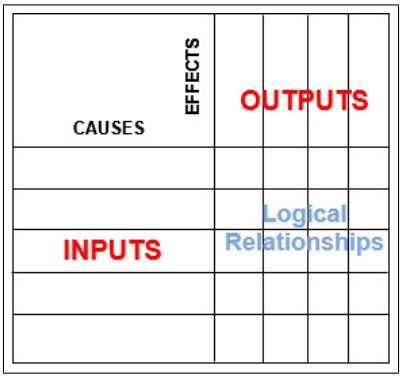

Figure 3. Structure of a Cause and Effect Matrix.

2.5 Translation of CEM to LTL properties

The IEC-61511.1 (2016) highlights two possible types of failures for a safety system: Spurious Trip Failure (ST) and Failure on Demand (FD). ST is a failure generated when the system is not required (does not have a cause), but the system acts as if there is a dangerous cause. And FD is a failure caused by a lack of action by the system, in the presence of a situation with a potential risk to the safety of the process. the safety properties in LTL are elaborated for each effect of the CEM, since the model checking verifies if properties are fulfilled or not by the system. In this way, Lázaro defined the following properties: Spurious Trip Freeness (STF) and Failure on Demand Freeness (FDF). The STF properties express that for all possible states reached by the system there will never be an effect without the motivating cause. FDF expresses that in all possible states, the system will never be demanded (occurrence of a cause) and will not act (no effect).

These properties can be systematically extracted from the CEM through the following LTL formulas:

$$STF = G\neg(probe \wedge \neg cause \wedge effect) \quad (1)$$

$$FDF = G\neg(probe \wedge cause \wedge \neg effect) \quad (2)$$

$$cause = [(effect' \vee (logical\_activate\_effect)) \wedge \neg(logical\_deactivate\_effect)] \quad (3)$$

Where:

- G is the "Globally" LTL temporal combiner, which asserts that all states in the future satisfy a given proposition (Bérard *et al.*, 2013);

- $\neg$, $\wedge$ and $\vee$ are the boolean combinators of negation, conjunction and disjunction, respectively;

- *probe* is the checking state in the BLD scan cycle, after updating the inputs and obtaining the outputs, as explained in section 4.

The causes of the CEM are expressed by logics that use the symbology of activation and deactivation of the effect. The updated value for each cause is determined using Eq. (1). The causes are related to the value of the effect in the previous time step (effect').

3. ALARM AND SEAL LOGIC

The seal logic integrated with alarm aims to retain the alarm signal until it is recognized by the operator, even if the cause of the alarm has already been normalized. In this way, the alarm and seal logic aims to ensure that the PLC continues to signal an alarm until it is notified of the alarm acknowledgment by the operator at the console, in order to enhance the safety and reliability of the system.

3.1 Model checking application

The model checking chain shown in Fig. 2 is used, at the conceptual design stage, to validate the specification of the alarm and seal logic of the CMS. The first steps are the definition of the BLD and the elaboration of the CEM. The BLD represents the behavior of the alarm and seal logic and the CEM represents the requirements that must be fulfilled by this control logic.

3.2 Cause and Effect Matriz

The CEM shown in Tab. 1 was built for the alarm and seal control logic. The symbology adopted for the development of the CEM is the one presented in subsection 2.4. The ENT cause represents the condition of an input variable in the PLC of the CMS. The REC cause represents the acknowledgment of the alarm by the operator. And the ALM effect represents the behavior of the alarm generated and sent by the PLC to the SCADA. The symbol A1+ represents the logical AND relationship between the causes that activate the ALM effect.

The Table 1 shows that the alarm should be activated when the ENT cause is activated and there is no REC acknowledgment. The alarm is deactivated when there is no ENT cause and there is REC acknowledgment. And symbol A2- relates through AND logic the causes that deactivate the ALM effect.

Table 1. Causes and effects matrix for alarm and seal logic.

Causes \ **Effect**	ALM
ENT	A1+, NA2-
REC	NA1+, A2-

3.3 Binary Logic Diagram

The MB technical team elaborated the specification of the PLC code through the Binary Logic Diagram. The Figure 4 presents an initial suggestion in BLD for the specification of the alarm and seal logic of the CMS. The BLD shows that the ALM signal assumes a high value when there is an abnormality in the system, with the variable ENT assuming a logic value of 1. The variable AUX is an auxiliary used to express the breaking of the seal in the logic. The ALM signal will only be deactivated (logical value 0) when the AUX signal assumes a high value and the ENT variable is reset (logical value 0). AUX signal assumes a high value when there is acknowledgment (REC in a high value).

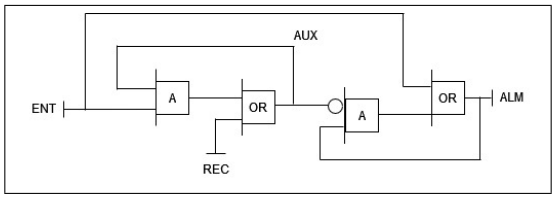

Figure 4. Initial BLD for alarm and seal logic.

3.4 Translation of CEM to LTL properties

Using Eq. (1) to (3), the safety requirements of CEM from Tab. 1 are systematically translated into safety properties: STF (Spurious Trip Freeness) and FDF (Failure on Demand Freeness) in LTL, as shown in Equations (4) to (7). Formal verification is performed in the PROBE state of the FIACRE model.

$$cause = [(ALM' \vee (ENT \wedge \neg REC)) \wedge \neg(\neg ENT \wedge REC)] \tag{4}$$
$$effect = ALM \tag{5}$$
$$STF = G\neg(PROBE \wedge \neg((ALM' \vee (ENT \wedge \neg REC)) \wedge \neg(\neg ENT \wedge REC)) \wedge ALM) \tag{6}$$
$$FDF = G\neg(PROBE \wedge ((ALM' \vee (ENT \wedge \neg REC)) \wedge \neg(\neg ENT \wedge REC)) \wedge \neg ALM) \tag{7}$$

3.5 Translation from BLD to FIACRE

Code 1 shows an extract of the FIACRE code for the BLD in Figure 4. A process with 4 states (UPDATE_INPUTS, UPDATES_OUTPUTS, PROBE and WAITING) is created to represent the BLD scan cycle. The BLD scan cycle defined by ISA 5.2 is divided into two steps: updating inputs and updating outputs. Initially, input variables are updated. Subsequently, there is the step of updating the output values, calculated by applying the logical operations specified in the BLD, which should occur in an orderly manner from left to right and top to bottom. Thus, the model in FIACRE represents a machine of four states, four transitions and six variables, where the states UPDATE_INPUTS and UPDATE_OUTPUTS represent the updates of input and output variables (after performing logical operations), respectively.

The PROBE state is the state where the model checking occurs and the WAITING state represents the restart of the BLD reading cycle. The variables ENT, REC, ALM and ALM_in represent the behavior of the variables: input, acknowledgment, alarm and the alarm in the previous cycle, respectively. The AUX variable is an auxiliary used to express the output signal of the first OR block, which returns as input to the first AND block, in Figure 4. And the AUX_in variable represents the value of the AUX variable in the previous reading cycle. In the transition between the UPDATE_INPUTS and UPDATE_OUTPUTS states: the ALM_in and AUX_in variables assume the values of ALM and AUX, in the previous reading cycle, respectively; and the ENT and REC variables can assume assume any input values from the system. In the transition between UPDATE_OUTPUTS and PROBE, the AUX and ALM output variables are updated according to the logic involved in the BLD. To ensure that the output values are always related to the input values at the end of each cycle, all transitions between states are defined as instantaneous (wait[0,0]), with the exception of the cycle restart transition.

Code 1. Main process of the FIACRE model for BLD of the figure 4.

```
process Diagram
    [portD_in: in bool, portD_out: out bool]
    is
    states UPDATE_INPUTS, UPDATES_OUTPUTS, PROBE, WAITING
    var ENT: bool := false,
    REC: bool := false,
    ALM_in: bool := false,
    ALM: bool := false,
    Aux: bool := false,
    Aux_in: bool := false
  init
    to UPDATE_INPUTS
  from UPDATE_INPUTS
    ALM_in := ALM;
    Aux_in := Aux;
    ENT := any;
    REC := any;
    wait[0,0];
    to UPDATES_OUTPUTS
  from UPDATES_OUTPUTS
    Aux := ((ENT and Aux_in) or REC);
    ALM := (ENT or ((not Aux) and ALM_in));
    wait[0,0];
    to PROBE
```

```
    from PROBE
      wait[0,0];
      to WAITING
    from WAITING
      to UPDATE_INPUTS
```

4. RESULTS

Following the methodology shown in Fig. 2, the model checker TINA/SELT receives as inputs: the TTS model, automatically translated from the FIACRE model through the FRAC tool; and the safety properties expressed in LTL formulas, by Eq. (6) and (7), in order to verify if the requirements defined in the Tab. 1 are fulfilled by the BLD shown in the Figure 4.

The result of the model checking for the alarm and seal logic of the CMS is shown in Tab. 2 and Figure 5. The Table 2 shows that the safety property Failure on Demand Freeness - FDF is satisfied by the BLD of the Figure 4. However, the property Spurious Trip Freeness - STF is not satisfied. The fact that the STF property is not satisfied by the system means that the ALM (alarm) effect is triggered in inappropriate situations (where there is no motivating cause).

The model checker explored 64 states and 112 transitions. The execution time is not significant, less than 1 ms, considering the simplicity of the system that has two inputs and one output.

Table 2. Result of the application of the methodology proposed by Lázaro *et al.* (2019) for the alarm and seal logic project.

Effect	**States**	**Transitions**	**Time**	**STF**	**FDF**
ALM	64	112	< 1ms	False	True

TINA/SELT provides a counterexample for the STF property, demonstrating the reason why the property is not satisfied by the system. This counterexample, in the form of a sequence of transitions from the TTS model, may not be easily understood by all members of the project team. However, some works propose methodologies for translating the counterexample to signal diagrams (a visual representation that is easy to understand), such as Marques *et al.* (2016).

Following the idea of signal diagram representation, the Fig. 5 shows the counterexample generated for the STF property. It shows that at time steps 8 and 10, the ALM effect is triggered just by the fact that the input signal ENT is activated, without taking into account whether the recognition signal REC is activated. According to the safety requirements, the alarm ALM should only be triggered when the signal ENT is activated and the signal REC is deactivated. Thus, the counterexample presents a possible scenario where the system does not fulfill the safety property STF, since the ALM effect is triggered inappropriately.

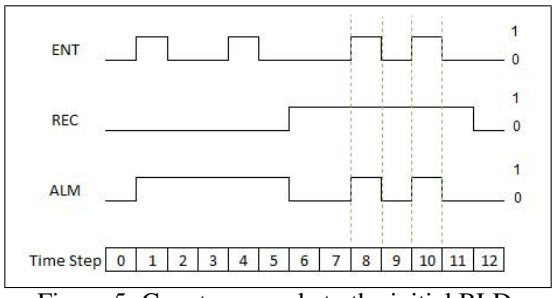

Figure 5. Counterexample to the initial BLD.

Based on the presented results, it is evident that the safety requirements defined by the CEM in Tab. 1 are not fulfilled by the BLD in Figure 4. The spurious activation error becomes a risk to the operation of the system because the operator cannot recognize repetitive alarms generated by inappropriate situations in the CMS, restricting the actions of intervention in the system, which may lead to possible incidents.

In order to correct this error, the BLD is modified to satify the safety requirements, as shown in Figure 6. In the modified BLD, ALM is activated only when there is ENT without REC, and it remains sealed as long as there is no REC or there is ENT. A new verification is performed, and the result is shown in Table 3.

The Tab. 3 shows that the safety properties STF and FDF are satisfied. In other words, the safety requirements defined by the CEM in Tab. 3 are fulfilled by the proposed BLD in Figure 6. For this simulation, TINA/SELT explored 32 states and 56 transitions in a computational time less than 1ms.

5. CONCLUSION

This article aimed to extend the validation of the a MC-based methodology for the CMS development project of the Brazilian Navy, proposed by Lázaro *et al.* (2019), through the application of methodology, with a focus on the conceptual

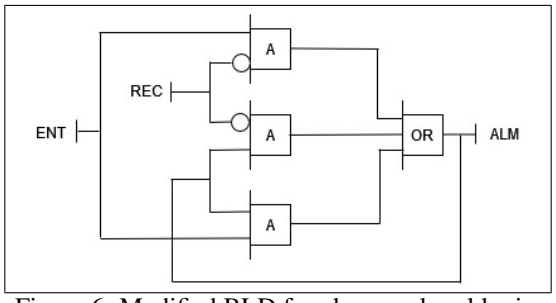

Figure 6. Modified BLD for alarm and seal logic.

Table 3. Result for modified BLD.

Effect	States	Transitions	Time	STF	FDF
ALM	32	56	< 1ms	True	True

design stage (BLD), to a real case of the CMS project: elaboration of the specification of the alarm and seal logic. This logic aims to retain the alarm signal generated by the PLC, in the face of an inappropriate situation, until the system operator acknowledges the alarm. A safety error associated with this logic can lead to possible incidents, with consequent risks to the life of the crew, material damage to the ship and environmental risks.

Initially, the methodology was applied to verify if the Binary Logic Diagram (BLD) suggested by the technical team of MB fulfilled the safety requirements expressed by the Cause and Effect Matrix (CEM). The formal verification showed that the initial BLD satisfies the safety property FDF (Failure on Demand Freeness) and does not satisfy the safety property STF (Spurious Trip Freeness). Furthermore, the model checker TINA/SELT presented a counterexample, from which it was possible to make corrections in the BLD. As a result, the BLD was corrected, and a new verification demonstrated that the modified BLD satisfies both the STF and FDF properties.

Therefore, the result of the case study demonstrates that the methodology is capable of validating the PLC code specifications of the CMS project through an automatic and exhaustive formal verification of the safety properties STF and FDF, as defined in the IEC 61511-1 standard. Additionally, the methodology can identify errors in the BLD. In this case, a counterexample is presented by the model checker, which helps the MB technical team in correcting the error in the initial stage of the project, where the cost is cheaper than in the other stages. Thus, increasing the reliability of the project and reducing the risk of possible incidents during the operation of the CMS. Thus, the methodology increases the reliability of the CMS project and reduces the risk of possible incidents during system operation.

6. ACKNOWLEDGEMENTS

The authors are grateful to Universidade Federal de Santa Catarina (UFSC) for access to its laboratory facilities and resources, and Instituto de Pesquisa da Marinha (IPqM) for providing the research data, which were essential to conduct the study.

7. REFERENCES

Adiego, B.F., Darvas, D., Viñuela, E.B., Tournier, J.C., Bliudze, S., Blech, J.O. and Suárez, V.M.G., 2015. "Applying model checking to industrial-sized plc programs". *IEEE Transactions on Industrial Informatics*, Vol. 11, No. 6, pp. 1400–1410.

Bérard, B., Bidoit, M., Finkel, A., Laroussinie, F., Petit, A., Petrucci, L. and Schnoebelen, P., 2013. *Systems and software verification: model-checking techniques and tools*. Springer Science & Business Media.

Berthomieu, B., Bodeveix, J.P., Farail, P., Filali, M., Garavel, H., Gaufillet, P., Lang, F. and Vernadat, F., 2008. "Fiacre: an intermediate language for model verification in the topcased environment". In *4th European Congress ERTS Embedded Real Time Software (ERTS 2008)*. p. 8p.

Berthomieu, B. and Vernadat, F., 2006. "Time petri nets analysis with tina." In *QEST*. Vol. 6, pp. 123–124.

Chadwick, S., James, P., Roggenbach, M. and Wetner, T., 2018. "Formal methods for industrial interlocking verification". In *2018 International Conference on Intelligent Rail Transportation (ICIRT)*. IEEE, pp. 1–5.

Clarke, E.M., 1997. "Model checking". In *Foundations of Software Technology and Theoretical Computer Science: 17th Conference Kharagpur, India, December 18–20, 1997 Proceedings 17*. Springer, pp. 54–56.

Farines, J.M., de Queiroz, M.H., da Rocha, V.G., Carpes, A.M.M., Vernadat, F. and Crégut, X., 2011. "A model-driven engineering approach to formal verification of plc programs". In *ETFA2011*. IEEE, pp. 1–8.

Gall, H., 2008. "Functional safety iec 61508/iec 61511 the impact to certification and the user". In *2008 IEEE/ACS*

International Conference on Computer Systems and Applications. IEEE, pp. 1027–1031.

Gergely, E.I., Coroiu, L. and Popentiu-Vladicescu, F., 2011. "Methods for validation of plc systems". *Journal of Computer Science and Control Systems*, Vol. 4, No. 1, p. 47.

IEC-61131.3, 2003. "Programmable controllers–part 3: Programming languages". International Electrotechnical Commission.

IEC-61508, 2010. "Functional safety of electrical/electronic/programmable electronic safety-related systems". International Electrotechnical Commission.

IEC-61511, 2023. "Functional safety - safety instrumented systems for the process industry". International Electrotechnical Commission.

IEC-61511.1, 2016. "Functional safety - safety instrumented systems for the process industry sector - part 1: Framework, definitions, system, hardware and application programming requirements." International Electrotechnical Commission.

IEC-62881, 2018. "Cause and effect matrix". International Electrotechnical Commission.

ISA-5.2, 1992. "Binary logic diagrams for process operation". International Society of Automation.

Lázaro, F.S., De Queiroz, M.H. and Farines, J.M., 2019. "Metodologia para desenvolvimento assistido por model checking de sistemas de controle e monitoração de navios". In *Congresso Brasileiro de Automática-CBA*. Vol. 1.

Liggesmeyer, P., Rothfelder, M., Rettelbach, M. and Ackermann, T., 1998. "Qualitätssicherung software-basierter technischer systeme–problembereiche und lösungsansätze". *Informatik-Spektrum*, Vol. 21, pp. 249–258.

Marques, L.G.P.C., de Queiroz, M.H. and Farines, J.M., 2016. "Improving a design methodology of synthesizable vhdl with formal verification". In *2016 IEEE 7th Latin American Symposium on Circuits & Systems (LASCAS)*. IEEE, pp. 51–54.

Pakonen, A. and Björkman, K., 2017. "Model checking as a protective method against spurious actuation of industrial control systems". In *27th European Safety and Reliability Conference, ESREL 2017*. CRC Press, pp. 3189–3196.

Reis, L.P.E. *et al.*, 2018. "Verificação formal de sistemas instrumentados de segurança na indústria de petróleo e gás natural".

Wing, J.M., 1990. "A specifier's introduction to formal methods". *Computer*, Vol. 23, No. 9, pp. 8–22.

8. RESPONSIBILITY NOTICE

The authors are solely responsible for the printed material included in this paper.