Formal Safety Assessment Methods in Olkiluoto 1&2 NPP I&C Renewal Project DIMA

Antti Pakonen, Lauri Tuominen, Mauri Viitasalo, Pekka Nuutinen

This is the submitted version. Please cite the final published version:

Pakonen, A., Tuominen, L., Viitasalo, M., Nuutinen, P. Formal Safety Assessment Methods in Olkiluoto 1&2 NPP I&C Renewal Project DIMA. 14th Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies (NPIC & HMIT 2025), June 15-18, 2025, Chicago, IL, USA, pp. 1774-1783.

DOI: 10.13182/NPICHMIT25-46672

Formal Safety Assessment Methods in Olkiluoto 1&2 NPP I&C Renewal Project DIMA

Antti Pakonen[1,*], Lauri Tuominen[2], Mauri Viitasalo[2], Pekka Nuutinen[2]

[1]VTT Technical Research Centre of Finland Ltd., Espoo, Finland
[2]TVO, Olkiluoto, Finland

ABSTRACT

Formal languages and methods provide a rigorous and systematic framework for requirement specification and design. A key benefit is the ability to logically prove the correctness of design solutions. TVO operates three nuclear power plants in Olkiluoto, Finland. Units 1 and 2 are BWR type reactors in use since 1979 and 1982. In 2020, TVO started the digital I&C lifetime management project DIMA, where selected I&C systems are renewed using mostly software-based technology. In DIMA, VTT has used two different formal, computer-assisted methods to verify the design solutions. First, model checking was used to verify the functional architecture diagrams serving as design input, and then the detailed I&C logic diagrams developed by Westinghouse Electric Sweden. The analyses have revealed several issues related to spurious actuation, contradictory commands, frozen logic, and in general, incorrect response to inputs. Second, on the overall I&C architecture level, an ontology-based Defense-in-Depth (DiD) assessment method was used to perform interface analyses. The work succeeded in identifying violations of communication independence rules between DiD levels and safety classes. For each violation, TVO was then able to prove by their analyses that failure propagation could not cause unacceptable consequences at the plant level. In this paper, we introduce the formal methods used in the DIMA project, describe the scope of their application, and discuss the results.

Keywords: formal verification, model checking, ontology, defense-in-depth, digital I&C

1. INTRODUCTION

The term *"formal methods"* refers to the use languages based on algebra, logic and set theory, and with strict syntax and grammar rules [1,2]. Formal specifications enable rigorous and systematic frameworks for engineering [2]. Furthermore, they offer the possibility to prove the correctness of the design with respect to the specifications [1].

In nuclear power plants (NPP), many modern instrumentation and control (I&C) systems are digital, based on software or programmable logic devices [1]. Digital I&C offers many advantages over the earlier analogue systems, but the additional complexity has introduced safety concerns [3,4]. The complexity also means that commonly used verification methods (e.g., testing, simulation, or code inspections) cannot achieve perfect coverage. However, the adoption of formal verification methods has been slow.

The overall I&C architecture establishes the organization of the I&C systems, the interconnections of the systems, the allocation of I&C functions to the systems, and the design constraints [4]. The overall architecture needs to be designed with the same attention to detail as is used with the system of the highest

*antti.pakonen@vtt.fi

safety class contained within it [5]. Of particular concern is the Defence-in-Depth (DiD) concept, where the DiD levels shall be sufficiently independent, so that failures cannot propagate between the levels [5]. The need for constant re-evaluation of the architecture as new details about the design become available [3] underlines the need for advanced methods and tools for DiD property assessment.

In this paper, we discuss the use of formal languages and methods in the Olkiluoto NPP I&C renewal project DIMA. In chapter 2, we introduce the DIMA project. In chapter 3, we describe how VTT[1] has used model checking to verify I&C logics. In chapter 4, we then describe the use of ontology-based techniques for analyzing DiD properties related to the overall I&C architecture. Finally, we present our conclusions in chapter 5.

2. OLKILUOTO NPP I&C RENEWAL

The Olkiluoto NPP, operated by TVO[2], is located on the western coast of Finland. It consists of three units: OL1 and OL2 are boiling water reactors (BWR), in operation since 1978 and 1980, respectively, and each with a capacity 890 MW. OL3 is an EPR with a capacity of 1600 MW, in regular operation since 2023.

ICMA (I&C lifetime management) is a renewal program for the I&C of OL1 and OL2 units. A part of ICMA is the *DIMA* (Digital I&C lifetime management) project, started in 2020.

The I&C systems of OL1 and OL2 being renewed in DIMA are:

- System 461, the *turbine control system*.
- System 531, the *neutron flux measuring system*.
- System 535, the *reactor power control system*.
- System 537, the *feedwater control system*.

In addition, the process computer system and the I&C system operator interface are renewed.

Systems 461, 535 and 537 are primarily responsible for non-safety control functions. However, the original DiD concept of the Generation II plant constrains the design solutions in the overall I&C architecture, and Safety Class 3 (SC3) functions are also allocated to each system. (Finnish SC3 corresponds with Class 2 / Category B in IEC 61226 [6].)

System 531 consists of two subsystems, a hardwired Safety Class 2 (SC2) part belonging to the reactor protection system, and a programmable SC3 part. (SC2 is the highest Finnish classification applicable to I&C systems, corresponding with Class 1 / Category A [6].)

SC3 functionality in the systems is implemented with the AC160 platform, originally developed by ABB, but now owned by Westinghouse Electric Company, and forming a part of the Common Q platform. The systems are supplied by Westinghouse Electric Sweden (WSE).

One challenge in the digitalization of systems in DIMA is that the documentation of the existing analogue I&C systems focuses on the implementation. Technology-independent requirements for the functionality and classification of the new systems cannot all be directly reverse-engineered from the analogue solutions.

Another challenge for project delivery is that the new systems need to be installed during annual service outages. The brief time window does not leave room for comprehensive testing of the systems in situ — and it is especially hard to test accident scenarios — so it is important to use all measures available to ensure that the design is correct before installation.

[1]https://www.vttresearch.com/en
[2]https://www.tvo.fi/en/

3. MODEL CHECKING OF I&C LOGICS

3.1. Method

Model checking [7] is a formal verification method, where a tool called a *model checker* is used to verify that a system model satisfies stated formal properties. The analysis is exhaustive and results in either a logical proof that the property holds, or a counterexample describing an execution of the model where the property fails. Symbolic model checkers (such as, e.g., NuSMV [8] or nuXmv [9]) utilize Binary Decision Diagrams (BDD) for compact representation of sets of model states, avoiding the need for explicit state enumeration [10].

The formal properties – derived from functional requirements – are expressed using *temporal logic* languages like Linear Temporal Logic (LTL) or Computation Tree Logic (CTL) [7]. LTL and CTL use temporal operators like **G** ("always"), **X** ("next"), or **F** ("eventually"), and CTL also uses path quantifiers **A** ("for all executions") and **E** ("there exists an execution"). For example, the *reachability* [10] property $\mathbf{AG}\,\mathbf{EF}\,\phi$ means that starting in every state, there shall exist some execution that can eventually reach state ϕ. The *safety* [10] property $\mathbf{G}\,\neg\phi$ means that ϕ shall never occur, and the *liveness* [10] property $\mathbf{G}(\phi \rightarrow \mathbf{F}\,\psi)$ means that the occurrence of ϕ shall eventually lead to the occurrence of ψ. Property types that occur commonly in VTT's projects are listed in [11].

Although property specification is error-prone and hard [11], model checking is self-repairing in the sense that an error made by the analyst in either the model or the property is typically then revealed by the resulting counterexample. The analyst can then iterate the process until the property is proven true for the model, or the counterexample reveals a real issue in the original system.

VTT and the power company Fortum have jointly developed a graphical tool called MODCHK [12], a front-end for the NuSVM model checker (see Fig. 1). MODCHK supports the modelling of non-standard, vendor-specific function block diagrams, including logics with signal validity processing [12]. MODCHK then generates the input file for NuSMV, runs NuSMV, and visualizes the counterexamples by animating the function block diagram.

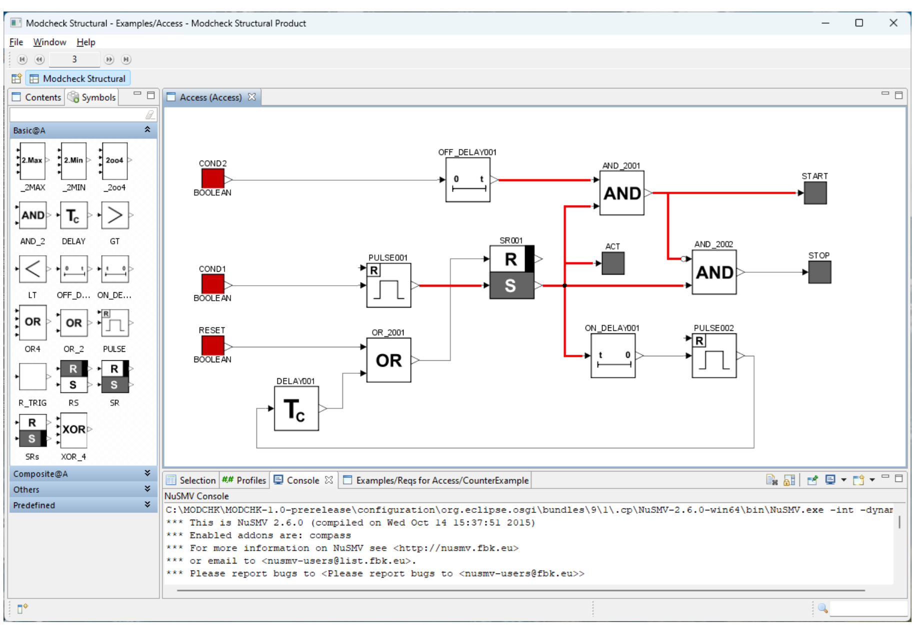

Figure 1: A simple example of counterexample animation in MODCHK (logic adopted from [11]).

3.2. Application in DIMA

The use of model checking in the DIMA project started with the verification of the *functional architecture* in 2022. The functional architecture defines and structures the functions that need to be performed, before assigning the functions to systems [4]. As the functional architecture is meant so serve as the input for the later design phases, high quality is essential. In DIMA, the functions were described using vendor-independent function block diagrams. VTT modelled the diagrams with MODCHK, formalized properties based on functional requirements, and verified the logics using NuSMV.

Starting in 2024, VTT has been verifying successive revisions of logic diagrams for systems 461, 531, 535, 537. The logic diagrams, authored by WSE, define the functions on a detailed level, using function blocks specific to the AC160 platform. VTT has modelled the diagrams with MODCHK and verified the logics using both NuSMV and nuXmv.

Since the analysis itself with tools such as NuSMV and nuXmv is typically very fast, and MODCHK makes it easy to work with graphical models, model checking was used iteratively to re-verify the DIMA designs as they evolved.

3.3. Results

VTT identified design issues in each verified system, both in the functional architecture and the later logic diagrams:

- Some issues were related to spurious actuation. With more conventional methods, it is easier to address the intended functionality, but with model checking, we can just as easily specify safety properties to exclude unintended actuations (e.g., $\mathbf{G}(\texttt{effect} \rightarrow \texttt{cause})$ [12]).
- Some issues were related to contradictory commands, which is also easy to exclude by specifying, e.g., $\mathbf{G}\,\neg(\texttt{open} \wedge \texttt{close})$.
- Some issues featured a signal permanently freezing to some value. We can easily detect such issues by specifying a CTL reachability property $\mathbf{AG}\,\mathbf{EF}\,\phi$, where the counterexample would feature the model being stuck in state $\neg\phi$.

Three issues were detected with nuXmv in functions that contained real number math operations too complex to be analyzed with NuSMV.

As a result of detecting the issues during the design phase, the deficiencies can be addressed and resolved well before the system implementation phase.

In addition to the DIMA project, VTT has also used model checking in the Olkiluoto 3 new-build project, the Loviisa 1&2 I&C renewal projects LARA and ELSA, and in the (cancelled) Hanhikivi 1 project. Practical examples of design issues VTT has identified can be found in [11,12,13].

3.4. Discussion

When using formal verification, it is important to consider and document the limitations of the methods and tools [2]. In DIMA, we checked if the way function blocks are connected on the diagrams can result in unwanted behavior. No assumptions were made about the correctness of the function block source codes, the generated application logic source code, or compiled machine instructions. The model is a simplified representation of the actual logic, and some aspects of the abstraction are preformed manually [12]. Although errors in property formalization are usually revealed through the resulting counterexample, it is difficult to guarantee that all relevant properties have been formalized.

Recent related studies on the use of model checking for verifying nuclear I&C system design include [14], [15], [16], [17] and [18].

4. ONTOLOGY-BASED INTERFACE ANALYSIS

In addition to individual I&C systems fulfilling their functional requirements, it is important for the overall I&C architecture to fulfil the Defense-in-Depth (DiD) principle. On one hand, the principle only works if the successive protection layers (DiD levels) are sufficiently independent of each other. On the other hand, complete independence of the DiD levels is practically impossible to achieve, being not just prohibitively costly, but also so complex that the difficulty in operation or maintenance can make such an architecture less safe than a properly optimized one [3]. Furthermore, in an I&C renewal project at a Generation II plant, the architectural solutions are necessarily constrained by the original DiD concept.

Independence between the DiD levels shall be based on adequate functional isolation, diversity, and physical separation [5]. To ensure that failures do not propagate between the levels, failure tolerance analyses shall consider the entire functional chain, including auxiliary systems [5].

One aspect of showing that the above requirements are met is to identify the interfaces where communication flows between DiD levels, and to prove that such communication does not enable failure propagation. A challenge for interface analysis in the DIMA project is that I&C functions of different DiD level (and safety class) have been allocated to the systems. Due to the functional chains across the systems, it is not sufficient to consider communication rule violations that occur on the system borders, but we also must analyze communication between the functions within each system.

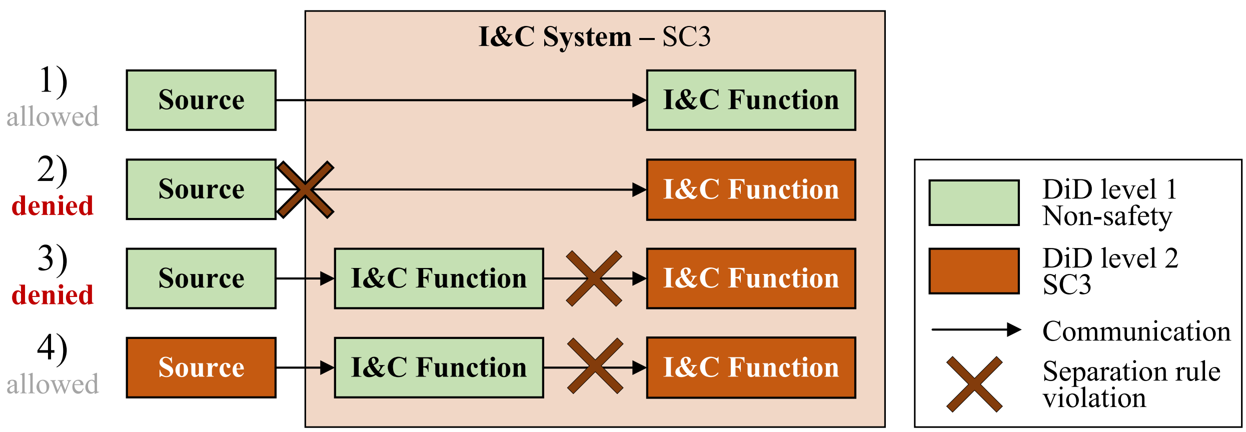

Figure 2: Some violations of communication independence can occur within an I&C system.

In Fig. 2, we show four simplified examples of functional chains for an exemplar I&C system of Safety Class 3 (SC3). In the DIMA interface analysis, we apply the following principles (following the numbering in Fig. 2):

1. If the entire functional chain is classified DiD level 1 (non-safety), a failure cannot propagate to DiD level 2 (SC3) functions. This communication path is therefore allowed.

2. If an external non-safety source communicates to a DiD level 2 (SC3) I&C function, then a failure can potentially propagate. This communication path is therefore **denied**.

3. Communication from a non-safety source to a DiD level 2 (SC3) I&C function can occur by way of a DiD level 1 (non-safety) function within the system. Although the violation is not apparent at the system interface, this communication path is still **denied**.

4. If the external source is classified DiD level 2 (SC3), then the entire functional chain is allocated to SC3 equipment, which is single failure tolerant. Even though communication passes through a DiD level 1 (non-safety) function, this communication path is still allowed.

A practical challenge in identifying the functional chains where communication independence is violated is that the analyst needs to combine pieces of information from different sources. For example, in Fig. 3, we depict a web of conceptual relationships related to the question: "are there interfaces where information from an external system flows to a function in the system of the interest, so that communication independence rules between DiD levels are violated?". In DIMA, to answer such a question, the analyst will need to combine information from the Safety concept, I&C architecture, and Signal list documents, along with the logic diagrams.

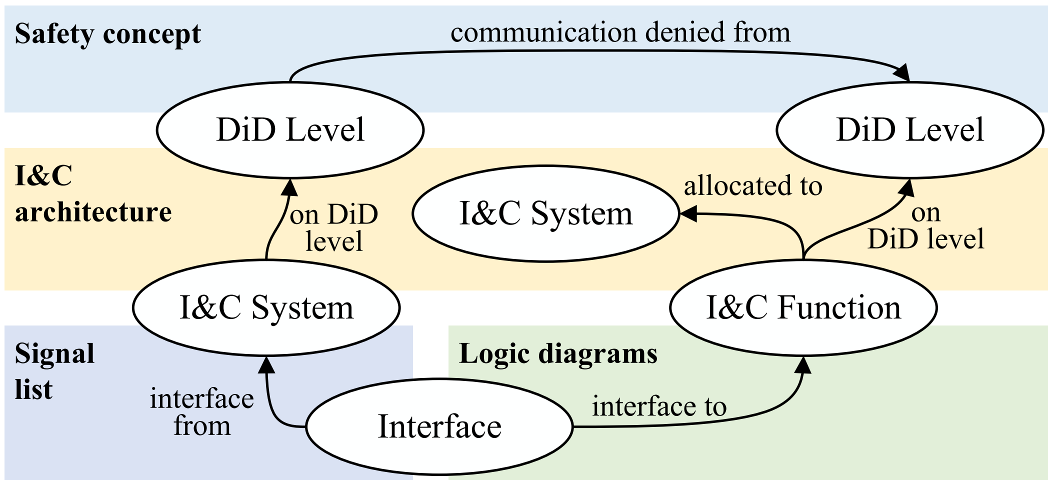

Figure 3: Answering queries over DiD properties requires information from different documents.

As the design evolves, even small changes (e.g., individual logic diagram wires) can affect the results. The architecture needs to be constantly re-evaluated, and proper tool support is therefore crucial [3].

4.1. Method

As is apparent from Fig. 3, answering queries over DiD properties can involve a "web" of concepts. The *Semantic Web* [19] represents an attempt to make the (often ill-structured) information on the Web easy for computers to search, combine and reason over by expressing the "meaning" of the information formally. By having an "understanding" of associations humans would find meaningful, computers can then merge, categorize, and synthesize information from scattered sources [20].

An *ontology* defines the domain concepts and their relationships, and OWL [21] is a language for writing formal ontologies that support automatic reasoning. A *knowledge base* contains statements about individuals of the classes the ontology defines, stored as directed, labeled graph — as RDF [21] triples. SPARQL [22] is a query language for RDF patterns, allowing complex queries.

As a simple example, we show below the SPARQL query that would correspond to Fig. 3.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?signal ?systemFrom ?didFrom ?functionTo ?didTo
  WHERE {
    ?interface rdf:type :Interface.
    ?interface :interfaceFrom ?systemFrom.
    ?systemFrom :associatedWithDidLevel  ?didFrom.
    ?interface :interfaceTo ?systemTo.
    ?interface :interfaceAssociatedWithFunction ?functionTo.
    ?functionTo :associatedWithDidLevel  ?didTo.
    ?didTo :communicationDeniedFromDiD ?didFrom.
    ?interface rdfs:comment ?l_signal.
  }
  ORDER BY ?didTo ?didFrom ?functionTo
```

Examples of DiD related queries where the reasoning supported by OWL is more complex (utilizing, e.g., taxonomies of properties, or transitive property types) can be found in [23].

VTT has developed an OWL ontology for representing engineering knowledge about nuclear overall I&C architectures [23]. Public case studies in [23] and [24] show how to specify SPARQL queries related to DiD properties such as physical separation, electrical isolation, communication independence, diversity, safety classification, and failure tolerance. The GraphDB [25] semantic repository has proved to be an efficient and scalable tool for storing the knowledge base and running the queries [24].

4.2. Application in DIMA

In DIMA, we used the ontology-based knowledge base to flexibly run queries about communication independence, to support interface analysis. The interface analysis consisted of three phases:

1. In functional dependency analysis, VTT collected and tabulated necessary information from the DIMA project documents (as shown in Fig. 3), constructed the knowledge base in GraphDB, executed SPARQL queries related to the separation rules, and collected the results indicating rule violations.

2. In failure propagation analysis, VTT used the logic diagrams to determine if failures could propagate through the interfaces that violate the separation rules, following the principles depicted in Fig. 2. We separately considered spurious actuation and failure on demand.

3. In failure consequence analysis, TVO experts determined if the propagation of failures could lead to consequences worse than the acceptance criteria set in Finnish regulations [5].

4.3. Results

The results for system 531 are still pending, but for each of the analyzed systems 461, 535, and 537:

1. VTT identified internal (between I&C functions within a system) and external interfaces where communication flows in a direction that violates a separation rule.

2. VTT showed that some of the rule violating interfaces allow a failure to propagate between DiD levels and safety classes in a way that violates failure tolerance requirements.

3. TVO successfully determined that the consequences of the possible failure propagation do not lead to consequences worse than the acceptance criteria.

The Finnish regulator STUK has already reviewed the interface analysis report for system 535, and accepted the pre-inspection documents for that system.

4.4. Discussion

The Semantic Web follows a rule-based approach, and the associations made between pieces of information derive from formal logic, not just statistical patterns. What nowadays is commonly referred to as Artificial Intelligence (machine learning, and often large language models) can deliver impressive results, but also “hallucinate” incorrect statements, which is not what we want in NPP safety assessment.

Successful implementations of ontology-based techniques in the nuclear domain are listed in [20]. As an example of a related I&C specific study, the DIAMOND ontology [26] was developed to support I&C component maintenance.

5. CONCLUSIONS

This paper described the successful use of formal methods in I&C software verification and overall I&C architecture DiD assessment in the Olkiluoto NPP I&C renewal project DIMA.

Model checking is an already well-established practice in the Finnish nuclear industry, having been used in all major I&C related activities since 2008. The DIMA project has been no exception, and we have again detected design issues not found in the utility's or supplier's own review processes, which have been based on more conventional methods.

In addition to improving safety, model checking can also reduce costs. The analysis itself is very fast, enabling rapid re-evaluation of design iterations or alternative design choices. Catching design issues early paves the way to faster project delivery. Crucially, as the installation needs to take place during annual service outages, any extra day spent on the installation due to unforeseen design issues not caught earlier can cost about a million euros.

For TVO, the number of detected issues also serves as a measure of the quality of the supplier's engineering processes. In the licensing process, the use of advanced verification methods serves as credible evidence in justifying the safety case to the regulator.

VTT's original motivation for developing the ontology-based DiD analysis approach was also motivated by practical considerations — we knew from practical experience that doing such analyses manually was tedious and slow. Semantic Web technologies help us flexibly specify complex queries, and have computers combine information from scattered sources based on logical rules.

In the ontology-based interface analysis for DIMA, we expected to find violations of communication independence rules. Some compromises are needed, as the overall I&C architecture is constrained by the original DiD concept. Nevertheless, formal analyses helped ensure that all violations are known, analyzed, and approved, also in the cases where communication crosses the DiD lines within the systems.

ACKNOWLEDGEMENTS

Both the model checking capabilities and the ontology-based DiD assessment method have been developed in public research programs — the Finnish Research Programme on Nuclear Power Plant Safety 2019-2022 (SAFIR2022), and the Finnish National Nuclear Safety and Waste Management Research Programme 2023-2028 (SAFER2028).

REFERENCES

[1] IAEA, "Introduction to Systems Engineering for the Instrumentation and Control of Nuclear Facilities," International Atomic Energy Agency, Nuclear Energy Series NR-T-2.14 (2022).

[2] Bel V, BASE, CNSC, CSN, KAERI, KINS, NSC, ONR, SSM, STUK and TIS, "Licensing of safety critical software for nuclear reactors – Common position of international nuclear regulators and authorised technical support organisations," Regulator Task Force on Safety Critical Software (2024).

[3] IAEA, "Approaches for overall instrumentation and control architectures of nuclear power plants," International Atomic Energy Agency, Nuclear Energy Series NP-T-2.1 (2018).

[4] EPRI,"Principles and Approaches for Developing Overall Instrumentation and Control Architectures that Support Acceptance in Multiple International Regulatory Environments," The Electric Power Research Institute, Technical Report 3002002953 (2014).

[5] STUK, "Safety design of a nuclear power plant," Radiation and Nuclear Safety Authority, YVL Guide B.1 (2019).

[6] WNA, "Safety Classification for I&C Systems in Nuclear Power Plants – Current Status & Difficulties", World Nuclear Association Report No. 2015/008 (2015).

[7] E. M. Clarke, O. Grumberg and D. Peled, *Model Checking*, MIT Press, Cambridge, MA (1999).

[8] A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani and A. Tacchella, "NuSMV 2: An opensource tool for symbolic model checking," *2002 International Conference on Computer Aided Verification (CAV)*, LNCS 2404, pp. 359-364 (2002).

[9] R. Cavada, A. Cimatti, M. Dorigatti, A. Griggio, A. Mariotti, A. Micheli, S. Mover, M. Roveri and S. Tonetta, "The nuXmv Symbolic Model Checker," in: A. Biere and R. Bloem (Eds.), *Computer Aided Verification*, Springer International Publishing, Cham, pp. 334-342 (2014).

[10] B. Bérard, M. Bidoit, A. Finkel, F. Laroussinie, A. Petit, L. Petrucci, P. Schoebelen and P. McKenzie, *Systems and Software Verification — Model-Checking Techniques and Tool*s, Springer, NY (2001).

[11] A. Pakonen, I. Buzhinsky and V. Vyatkin, "Evaluation of visual property specification languages based on practical model-checking experience," *Journal of Systems and Software*, **216**, 112153 (2024).

[12] A. Pakonen, I. Buzhinsky and K. Björkman, "Model checking reveals design issues leading to spurious actuation of nuclear instrumentation and control systems," *Reliability Engineering & System Safety*, **205**, 107237 (2021).

[13] A. Pakonen, "Oops! Examples of I&C design issues detected with model checking," *International Symposium on Future I&C for Nuclear Power Plants (ISOFIC2021)*, Okoyama, Japan, Nov. 15-17 (2021).

[14] I. D. Lopez-Miguel, J.-C. Tournier and B. Fernandez, "PLCverif: Status of a formal verification tool for programmable logic controller," *18th International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS 2021)*, Shanghai, China, Oct. 14-22 (2021).

[15] T. Ausberger, K. Kubíček, and P. Medvecová, "Functionally-equivalent formalisation and automated model checking of Function Block Diagrams," *IEEE Access*, **13**, pp. 22197-22229 (2025).

[16] S. Gautham, A. V. Jayakumar, A. D. Rajagopala and C. Elks, "Accessible Formal Verification for Design Assurance of Model Based Critical I&C Systems," *12th Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies (NPIC&HMIT2021)*, Providence, RI, USA, June 14-17, pp.802-811 (2021).

[17] A. Wakankar, A. Kabra, A. K. Bhattacharjee and G. Karmakar, "Architectural model driven dependability analysis of computer based safety system in nuclear power plant," *Nuclear Engineering and Technology*, **51**(2), pp.463-478 (2019).

[18] V. Kumar, K. C. Mishra, P. Singh, A. N. Hati, M. R. Mamdikar, L. K. Singh and R. N. R. Parida, "Reliability analysis and safety model checking of safety-critical and control systems: A case study of NPP control system," *Annals of Nuclear Energy*, **166**, 108813 (2022).

[19] N. Shadbolt, T. Berners-Lee and W. Hall, "The Semantic Web revisited," *IEEE Intelligent Systems*, **21**(3), pp. 96–101 (2006).

[20] IAEA, "Exploring semantic technologies and their application to nuclear knowledge management," International Atomic Energy Agency, Nuclear Energy Series NG-T-6.15 (2021).

[21] "OWL 2 Web Ontology Language Document Overview (Second Edition)," W3C Recommendation, The World Wide Web Consortium, https://www.w3.org/TR/owl2-overview/ (2012).

[22] "SPARQL 1.1 Query Language," W3C Recommendation, The World Wide Web Consortium, https://www.w3.org/TR/sparql11-query/ (2013).

[23] A. Pakonen and T. Mätäsniemi, "Ontology-based approach for analyzing nuclear overall I&C architectures," *47th Annual Conference of the IEEE Industrial Electronics Society (IECON 2021)*, Toronto, Canada, Oct. 13-16 (2021).

[24] K. Björkman and A. Pakonen, "Analyzing defense-in-depth properties of nuclear power plant instrumentation and control system architectures using ontologies," *13th Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies (NPIC & HMIT 2023)*, Knoxville, TN, USA, July 15-20, pp. 1590-1599 (2023).

[25] "GraphDB Semantic Graph Database," Ontotext, https://graphdb.ontotext.com/ (2022).

[26] A. Al Rahsdan, J. Browning and C. Ritter, "Data integration aggregated model and ontology for nuclear deployment (DIAMOND): Preliminary model and ontology," Idaho National Laboratory, Technical Report INL/EXT-19-55610, 2019.