Safety Assurance of Software and Machine Learning Development for Nuclear Instrumentation and Controls

1st Sophia Zhu
Idaho National Laboratory
Idaho Falls, ID, USA
sophiazhu16@gmail.com

2nd Congjian Wang
Idaho National Laboratory
Idaho Falls, ID, USA
Congjian.Wang@inl.gov

3rd Jisuk Kim
Idaho National Laboratory
Idaho Falls, ID, USA
Jisuk.Kim@inl.gov

***Abstract*—Digital instrumentation and control (DI&C) systems monitor and control parameters in nuclear power plants. Ensuring their safety is a critical part of ensuring overall plant safety. Nuclear power plant licensing generates thousands of safety documents that could be organized more effectively using a safety assurance case (SAC). We conducted a literature survey of SACs and created a SAC framework for DI&C software using Goal Structuring Notation (GSN). This framework focuses on four software development processes: management & assurance, pre-developed software (PDS) qualification, the Software Development Life Cycle (SDLC), and the Machine Learning Development Life Cycle (MLDLC). We organized our framework using a novel level structure that can be applied to other SACs to improve their clarity. Finally, we demonstrate how our framework can be incorporated as part of a SAC for a larger reactor system.**

***Index Terms*—digital instrumentation and control, safety assurance case, Goal Structuring Notation, software assurance, machine learning assurance**

I. Introduction

Instrumentation and control (I&C) systems are used in nuclear power plants to monitor parameters and regulate them within an acceptable range. Increasingly, analog I&C systems are being replaced with digital I&C (DI&C) systems to improve safety and reliability and to facilitate equipment monitoring and testing. Assurance of DI&C systems, especially of those that are safety-critical, plays a crucial role in ensuring the overall safety of the nuclear power plant. Major publishers of nuclear DI&C software standards include the Institute of Electrical and Electronics Engineers (IEEE) and the International Electrotechnical Commission (IEC). Currently, the U.S. Nuclear Regulatory Commission (NRC) regulates DI&C software through multiple documents, including the Standard Review Plan (SRP), Regulatory Guides, General Design Criteria, and select IEEE standards [1]. The licensing process of a nuclear reactor generates thousands of documents, making it difficult to obtain an overview of the safety argument as a whole.

In this paper, we chose to organize those documents using a safety assurance case (SAC). SACs, sometimes referred to simply as safety cases, are structured arguments, often conveyed in a diagram format, that are used to verify the capabilities of safety-critical systems to avoid failures [2]. A SAC diagram for DI&C software organizes relevant safety requirements and documents, communicating the safety argument by elucidating the relationships between different parts of the argument.

SACs are used in a variety of fields, including nuclear power, software, artificial intelligence (AI) & machine learning (ML), ground & aerospace vehicles, and medical devices [2], [3]. In the UK, safety cases became a legal requirement for licensing nuclear facilities in the Nuclear Installations Act of 1965 [4]. Accidents like Three Mile Island in 1972 and Piper Alpha in 1988 spurred further development of SACs and their introduction to other industries [3]. In the 2010s, the U.S. Food and Drug Administration conducted a pilot project to mandate SACs for infusion pumps. The results demonstrated that SACs effectively reduce regulatory burdens and convey the manufacturer's understanding of safety concerns [5], [6].

Research on the safety assurance of ML is currently dominated by the field of autonomous vehicles, but there is ML safety research being done in the nuclear field as well. Increasing the autonomous abilities of reactors can reduce the likelihood of accidents and core damage by minimizing opportunities for human error [7]. However, ML models can also introduce uncertainty and complexity, which poses barriers to regulatory acceptance [7]. As ML technologies improve, their application within the nuclear industry will likely increase, and thus their presence in SACs.

There are multiple ways of creating a SAC. Two widely used diagram formats are Goal Structuring Notation (GSN) and Claim–Argument–Evidence (CAE) [2]. The claims, arguments, and evidence of CAE correspond to the GSN core elements of goals, strategies, and solutions, respectively [2]. GSN also has supporting elements of context, assumptions, and justifications to clarify goals and strategies, as well as additional symbols [8], which are illustrated in Fig. 1. We

This work of authorship (INL/CON-25-87368) was prepared as an account of work sponsored by Idaho National Laboratory (under Contract DE-AC07-05ID14517), an agency of the U.S. Government.

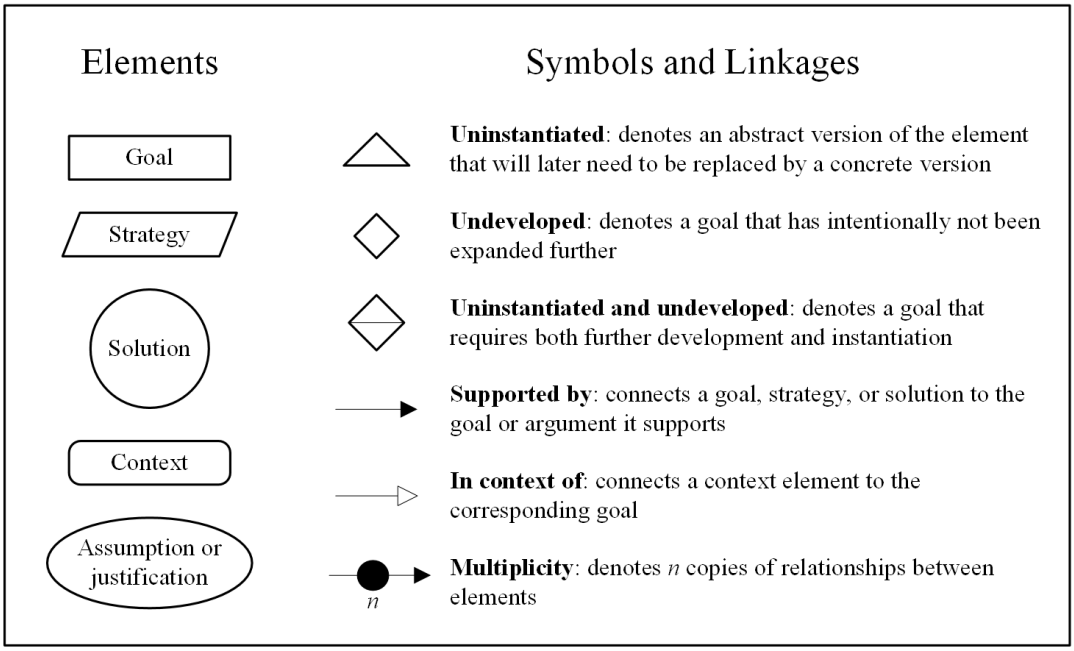

Fig. 1. GSN Symbology from the GSN Community Standard [8]

found that these extra elements and symbols were useful in our framework, so we decided to use GSN for it.

The DI&C section of the SRP [1] refers to two types of requirements—functional and process—that must both be met. Functional requirements ensure a safety system is able to perform its required functions, while process requirements ensure the system is built in a manner that contributes to safety and quality. The NRC [1] notes that due to the complex nature of software, running test cases is insufficient for guaranteeing the functional requirements are met, so ensuring sufficient quality of the software development process is particularly important. Hence, our GSN framework will focus on the software process requirements.

II. Proposed Assurance Case Framework

We conducted a literature survey on SACs and used the information to develop a SAC framework for software development processes.

A. Framework Overview

We have incorporated a few methods to assist readers in following the framework, which spans Figs. 2–4. First, we have labeled each element with a code, such as G1. The letters G, S, Sn, and C stand for "goal," "strategy," "solution," and "context," respectively. We did not use assumption or justification elements. The numbers are simply assigned in the order that the elements appear.

We have also introduced a new organizational method to supplement GSN in which every two rows are grouped as a level. The first row in each level contains goal elements, while the second row contains either strategy or solution elements. Context, assumption, and justification elements may be present in either row. Since the framework is divided into multiple parts due to space limitations, this level structure is intended to help readers understand the relative locations of the parts of the diagram.

Fig. 2 depicts an overview of the framework. Level 1 contains the main goal or top goal: "The DI&C component's software is safe for operating in the given conditions" (G1). This is demonstrated through two strategies: satisfying safety requirements (S1) and mitigating hazards (S2). This method of breaking down the top goal is common in SACs [3], [9], [10]. It results in the Level 2 goals of "All safety requirements are satisfied" (G2) and "The risks associated with hazards are sufficiently mitigated" (G3). In Level 3, G2 is split into functional and process requirements. To recap from the introduction (Section I), functional requirements are requirements for the software's functionalities, and process requirements are requirements for the software development process.

As stated in the introduction, we will focus on the safety process requirements (G5, highlighted). G3 and G4 are left undeveloped and are briefly addressed in Section III. We identified four main process safety requirements (G6-G9) and list these in Level 4B [1], [7], [9], [11], [12]. Management & assurance techniques (G6) and PDS qualification (G7) are discussed in Section II-B, while the Software Development Life Cycle (SDLC) (G8) and ML Development Life Cycle (MLDLC) (G9) are discussed in Sections II-C and II-D.

B. Management & Assurance and PDS (G6, G7)

Management and assurance techniques (G6) are activities that support software development. They include configuration management [1], [11]; quality assurance [1]; team compe-

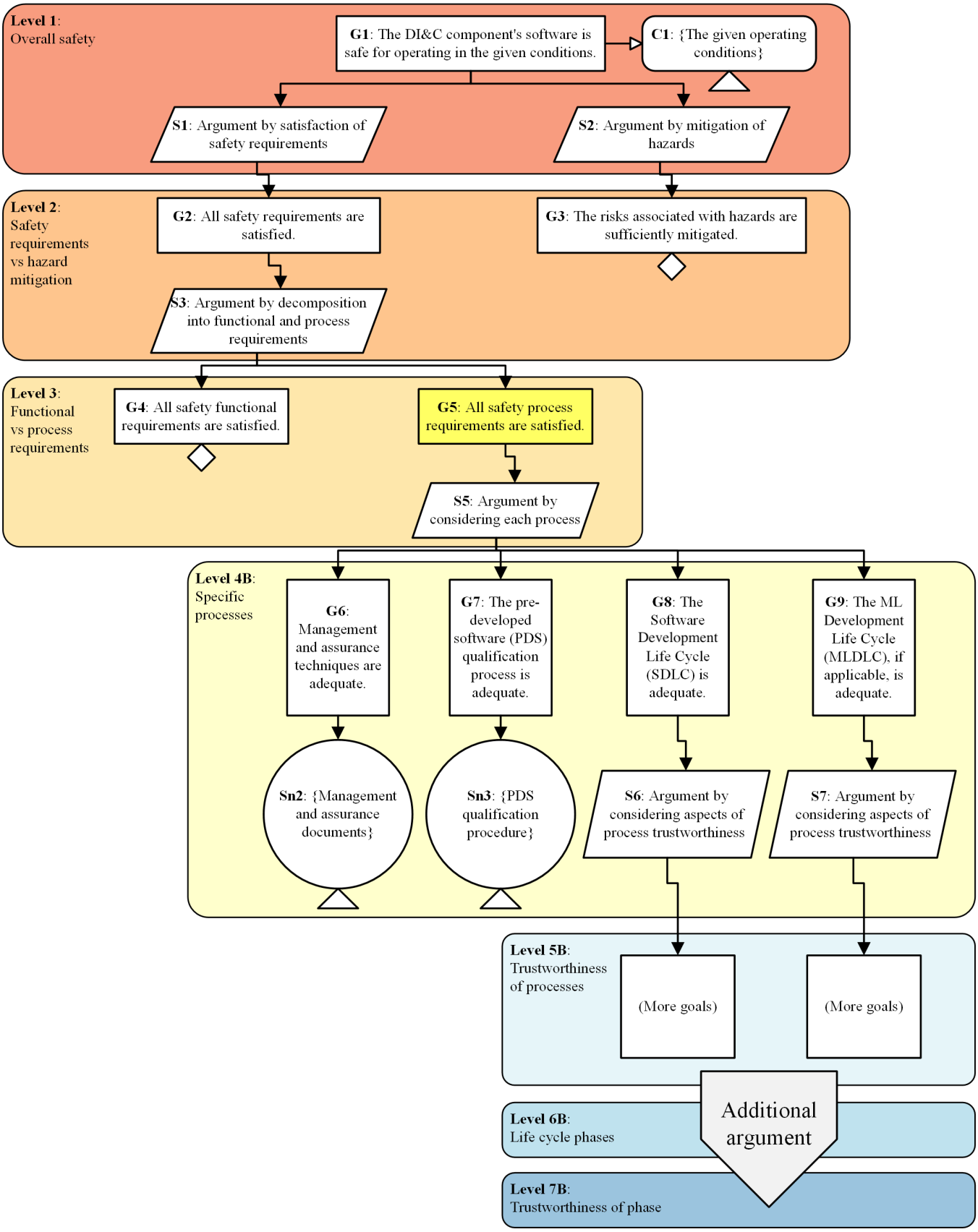

Fig. 2. Framework overview

tency [9]; standards compliance [1]; verification and validation (V&V), reviews, and audits [1], [11]; and risk management [11]. V&V is also addressed in the development life cycles (G8 and G9) and is a required part of each development phase (Section II-D).

PDS qualification (G7) is another process that support the development life cycles, but it is addressed separately from the management and assurance techniques in this framework due to its particular relevance to software. PDS refers to software not developed specifically for the component in question (commercial or proprietary software) being considered for use as a development tool or as part of the DI&C software [1], [12]. All PDS must be qualified prior to use to ensure it meets the same safety standards as other DI&C software.

Standards for PDS qualification are detailed in IEC 60880 [12]. In contrast, the NRC [1] does not enforce specific process requirements; instead, it recommends using EPRI TR-106439, "Guideline on Evaluation and Acceptance of Commercial-Grade Digital Equipment for Nuclear Safety Applications" [13] as guidance.

C. Development Life Cycles (G8, G9)

Fig. 3 illustrates the breakdown of the SDLC (G8) and MLDLC (G9). Because the use of ML in nuclear I&C is still in the early research stages, most implementations of this framework will not require the MLDLC branch.

Both the SDLC and MLDLC are a series of phases with defined activities for each phase. There are a variety of established SDLC models. The waterfall model [14] is the most traditional and includes the following phases, according to IEEE Standard 7-4.3.2 [11]:

1) Conceptual design
2) System requirements
3) Detailed system design
4) Implementation
5) Testing
6) Installation and site acceptance testing
7) Operations and maintenance
8) Retirement

It is not uncommon to see slight variations in the specific phases of the waterfall model. Other models fundamentally different from the waterfall model include the incremental [14], agile [15], and spiral [16] models.

Though ML processes are less established than software development, MLDLC models do exist, such as the Assurance of ML for Use in Autonomous Systems (AMLAS) developed by Paterson et al. [17]. AMLAS includes the following phases:

1) ML safety assurance scoping
2) ML requirements
3) Data management
4) Model learning
5) Model verification
6) Model deployment

Level 5B (Fig. 3) is used to establish the trustworthiness of the SDLC and MLDLC models. One aspect of trustworthiness is that the models must be chosen with consideration for their benefits and limitations (G10 and G12). For instance, the dynamic and fast-paced nature of the agile model makes it unsuitable for meeting the strict safety requirements of the nuclear industry. The NRC Standard Review Plan [1] does not require any specific SDLC model, but it does require that the life cycle is planned adequately and includes regular verification. As of this time, the NRC has not yet provided guidance on ML development processes.

The other aspect of model trustworthiness is that the chosen model must be followed correctly (G11 and G13). We have incorporated flexibility into our framework for different SDLC and MLDLC models by creating a framework section to assure the trustworthiness of any life cycle phase. As shown in Level 6B (Fig. 3), every phase of the chosen models is considered in a separate instance of G14. The trustworthiness of the phase is elaborated on in Section II-D.

D. Life Cycle Phases (G14)

Fig. 4 demonstrates a framework that is applied to every life cycle phase for the SDLC and MLDLC. In Level 7B, the trustworthiness of each phase is ensured: it must be fully planned (G15) and executed (G16) and must go through the V&V process (G17) [18]. The context element C4 will need to be instantiated with relevant standards. Some general standards include:

- All necessary stakeholders are involved during the planning and execution of the phase [18].
- V&V is performed by an independent and technically competent group with sufficient time and resources [11].
- All identified issues are resolved and the corrections verified [19].

The execution outputs (Sn8) will vary depending on the phase—for instance, the requirements phase could produce a requirements specification and a requirements analysis, while the operation phase could produce a maintenance log.

Until the reactor is decommissioned, there will be phases in the life cycles that will occur in the future. These phases must not be left out of the SAC, as there should be plans for each phase created near the beginning of the life cycle. SACs are most effective as living documents; additional information should be regularly added as development progresses [20].

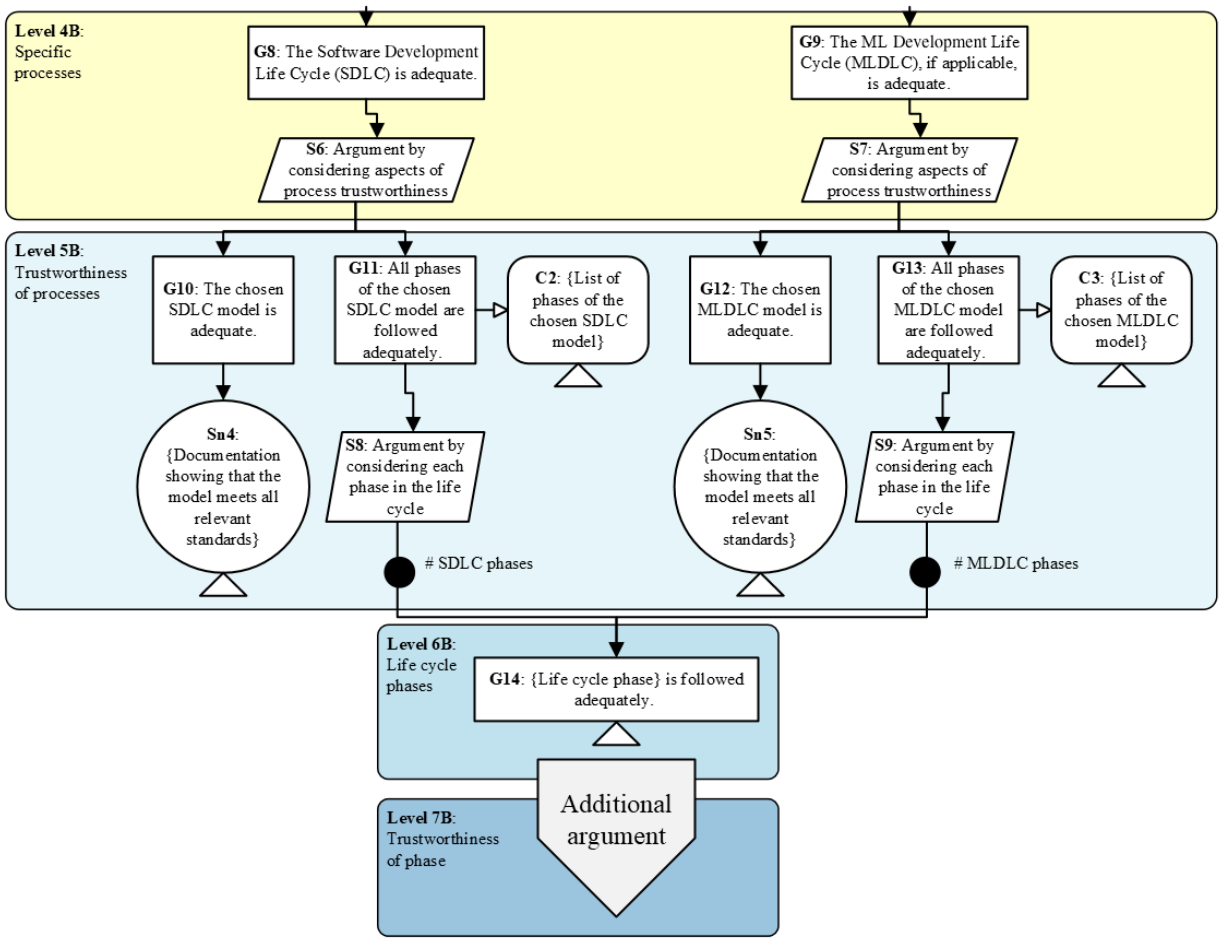

Fig. 3. SDLC and MLDLC in the framework

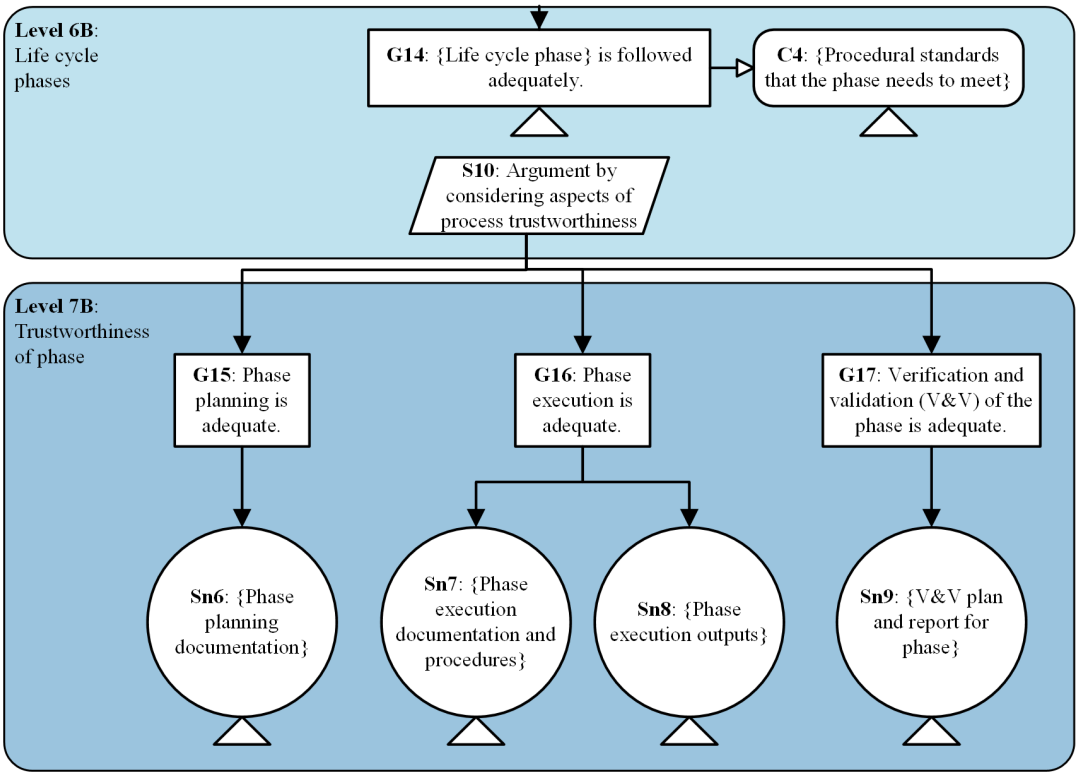

Fig. 4. Life cycle phase trustworthiness in framework

III. Discussion

In this section, we will briefly discuss the functional safety requirements (G4) and hazard mitigation (G3) branches of the framework, which were left as undeveloped elements. We will also examine how this framework relates to the safety assurance of larger reactor systems that include DI&C components.

A. Functional Safety Requirements (G4)

Element G4 states, "All safety functional requirements are satisfied" (Fig. 2). Real-time performance requirements, which the NRC addresses in BTP 7-21 [21], belong here. There are also component-specific requirements. For example, the component may be required to perform specific calculations or to control specific components. IEC 60880 [12] contains detailed coding standards, for example how to choose a programming language, which could also be included under G4.

B. Hazard Mitigation (G3)

G3 states, "The risks associated with hazards are sufficiently mitigated." This goal, like G2 ("All safety requirements are satisfied"), can be split into functional and process requirements. The major requirements involved are self-testing and surveillance (functional), hazard analysis (process), and software security (both functional and process).

Software self-testing and surveillance activities are required by the NRC. These activities include automatic error checking, memory management, and central processing unit monitoring. They are discussed in Branch Technical Position 17, "Guidance on Self-Test and Surveillance Test Provisions" (BTP-17) [22], in the NRC Standard Review Plan. IEC 60880 [12] contains detailed self-testing and surveillance standards as well.

The hazard analysis process involves identifying hazards, evaluating their risks, and reducing the risks to an acceptable level if necessary. Methods for hazard analysis include preliminary hazard analysis, fault tree analysis, and failure mode and effects analysis (FMEA); these and other methods are referenced in IEEE 7-4.3.2. Annex D [11]. Identifying software failure modes and their effects, which is often achieved through FMEA, is required by the NRC in the SRP [23] as part of the probabilistic risk assessment for the reactor as a whole. The NRC [23] notes that software failure modes can be difficult to characterize, so it is necessary to check that the FMEA process is systematic and thorough.

Software security requirements are detailed in Regulatory Guide 1.152 [24]. A majority of this document addresses process requirements for each phase in the SDLC. The guide also includes some functional software security requirements, including the inclusion of access control and the lack of undocumented, malicious, or unwanted code. IEEE 7-4.3.2 [11] contains additional cybersecurity requirements. For instance, the inclusion of cybersecurity features into a safety system should not adversely impact its functionalities.

C. Safety of Reactor Systems (Outside the Framework)

The proposed SAC framework focuses on the software of a given DI&C component. It can be incorporated as part of a larger SAC for the reactor or for systems inside the reactor, such as the I&C system or the reactor protection system, as illustrated in Fig. 5.

For larger systems, interactions between components become especially important. Consideration of factors such as electromagnetic interference, data communications, diversity and defense-in-depth, independence, and common-cause failure is explicitly required by both the U.S. NRC [1] and Canadian Nuclear Safety Commission [25].

IV. Conclusion

In this paper, we have introduced a SAC framework for DI&C software, focusing on process requirements. Unlike other frameworks that we reviewed, our framework incorporates relevant NRC documents and accommodates different SDLC and MLDLC models. The level structure we developed for our framework has the potential to be applied to other SACs to enhance their comprehensibility. As part of ongoing research, we aim to further develop the currently undeveloped branches of the framework and to apply it to the Advanced Power Reactor 1400's Core Protection Calculator System as a case study.

References

[1] Nuclear Regulatory Commission, "Appendix 7.0-A: Review Process for Digital Instrumentation and Control Systems," in *Standard Review Plan for the Review of Safety Analysis Reports for Nuclear Power Plants: LWR Edition*, NUREG-0800, Aug. 2016. [Online]. Available: https://www.nrc.gov/docs/ML1601/ML16019A085.pdf

[2] M. Sivakumar et al., "The Last Decade in Review: Tracing the Evolution of Safety Assurance Cases through a Comprehensive Bibliometric Analysis." arXiv, Nov. 13, 2023. doi: 10.48550/arXiv.2311.07495. [Online]. Available: http://arxiv.org/abs/2311.07495

[3] D. Rinehart, J. Knight, and J. Rowanhill, "Current Practices in Constructing and Evaluating Assurance Cases With Applications to Aviation," NASA, Jan. 2015. [Online]. Available: https://ntrs.nasa.gov/api/citations/20150002819/downloads/20150002819.pdf

[4] J. M. Brain, "Learning from experience – how can we produce a nuclear safety case to outlast the station?," in *9th IET International Conference on System Safety and Cyber Security* (2014), Oct. 2014, pp. 1–6. doi: 10.1049/cp.2014.0972. [Online]. Available: https://ieeexplore.ieee.org/document/7111727

[5] "Infusion Pump Improvement Initiative," *FDA*, Jun. 2024. [Online]. Available: https://www.fda.gov/medical-devices/infusion-pumps/infusion-pump-improvement-initiative

[6] F. Wu, "Introduction of Assurance Case Method and its Application in Regulatory Science," Silver Spring, MD: FDA, May 2019. [Online]. Available: https://www.fda.gov/media/125182/download

[7] H.-Y. Hsieh and P. Tsvetkov, "Advancements and challenges of machine learning and deep learning in autonomous control of nuclear reactors," *Annals of Nuclear Energy*, vol. 223, p. 111643, Dec. 2025, doi: 10.1016/j.anucene.2025.111643.

[8] *GSN Community Standard*, Version 3, The Assurance Case Working Group, England, May 2021. [Online]. Available: https://scsc.uk/index.php/gsn-standard

[9] I. Habli and T. Kelly, "Process and Product Certification Arguments: Getting the Balance Right," *SIGBED Rev.*, vol. 3, no. 4, pp. 1–8, Oct. 2006, doi: 10.1145/1183088.1183090. [Online]. Available: https://dl.acm.org/doi/10.1145/1183088.1183090

[10] K.-C. Kwon, J.-S. Lee, and E. Jee, "A Framework for the Safety Assurance of Safety Software in Nuclear Power Plants," Gyeongju, Korea, Nov. 2017. [Online]. Available: https://www.kns.org/files/int_paper/paper/ISOFIC_2017_10/ISOFIC-2017-Kee-Choon-KWON.pdf

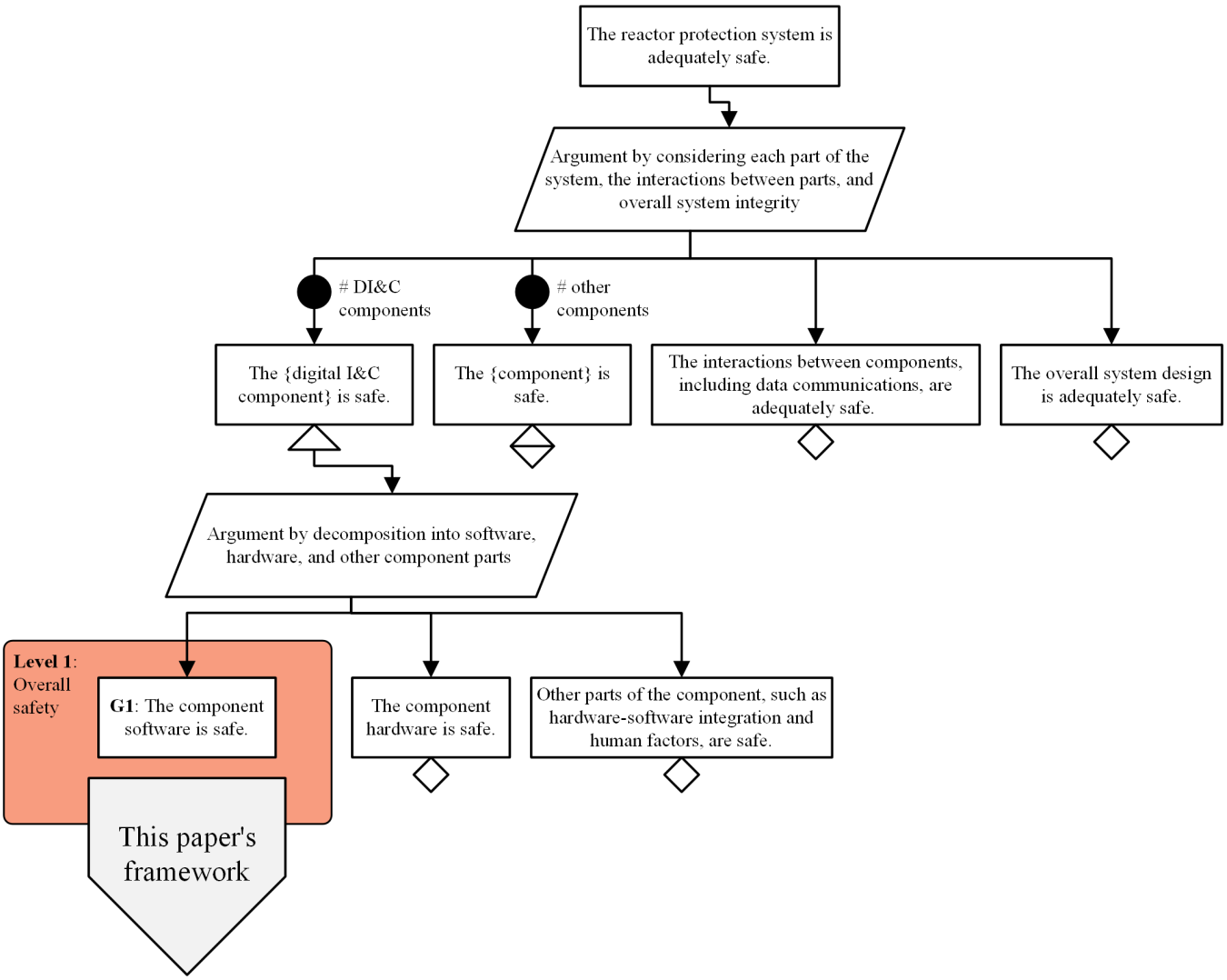

Fig. 5. Potential placement of this framework's top goal (highlighted) in the context of a larger SAC

[11] *IEEE Standard Criteria for Programmable Digital Devices in Safety Systems of Nuclear Power Generating Stations*, IEEE 7-4.3.2, Aug. 2016, doi: 10.1109/IEEESTD.2016.7552419.

[12] *Nuclear Power Plants - Instrumentation and Control Systems Important to Safety - Software Aspects for Computer-Based Systems Performing Category A Functions*, IEC 60880, 2006.

[13] *Guideline on Evaluation and Acceptance of Commercial-Grade Digital Equipment for Nuclear Safety Applications*, EPRI TR-106439, Palo Alto, CA, USA, Oct. 1996. [Online]. Available: https://www.epri.com/research/products/tr-106439

[14] A. M. Davis, E. H. Bersoff, and E. R. Comer, "A strategy for comparing alternative software development life cycle models," in *IEEE Transactions on Software Engineering*, vol. 14, no. 10, pp. 1453-1461, Oct. 1988, doi: 10.1109/32.6190.

[15] M. Lindvall et al., "Agile software development in large organizations," in Computer, vol. 37, no. 12, pp. 26-34, Dec. 2004, doi: 10.1109/MC.2004.231.

[16] B. W. Boehm, "A Spiral Model of Software Development and Enhancement," in Readings in *Human–Computer Interaction*, Elsevier, 1995, pp. 281–292, doi: 10.1016/B978-0-08-051574-8.50031-5.

[17] C. Paterson, R. Hawkins, C. Picardi, Y. Jia, R. Calinescu, and I. Habli, "Safety assurance of Machine Learning for autonomous systems," *Reliability Engineering & System Safety*, vol. 264, p. 111311, Dec. 2025, doi: 10.1016/j.ress.2025.111311. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0951832025005125

[18] R. Youngblood III, H. Everett, and H. Dezfuli, "Application of Objectives-Drive Assurance Cases to System Development in an Evolving Acquisition Model," Oct. 2022. [Online]. Available: https://www.osti.gov/biblio/2403507

[19] *Systems and software engineering – Software life cycle processes*, ISO/IEC/IEEE 12207, Nov. 2017, doi: 10.1109/IEEESTD.2017.8100771.

[20] Z. Langari and T. Maibaum, "Safety Cases: A Review of Challenges," in *Proceedings of the 1st International Workshop on Assurance Cases for Software-Intensive Systems*, 2013, pp. 1–6. [Online]. Available: https://dl.acm.org/doi/10.5555/2662398.2662400

[21] Nuclear Regulatory Commission, "Branch Technical Position 7-21: Guidance on Digital Computer Real-Time Performance," in *Standard Review Plan for the Review of Safety Analysis Reports for Nuclear Power Plants: LWR Edition*, NUREG-0800, Aug. 2016. [Online]. Available: https://www.nrc.gov/docs/ML1602/ML16020A036.pdf

[22] Nuclear Regulatory Commission, "Branch Technical Position 7-17: Guidance on Self-Test and Surveillance Test Provisions," in *Standard Review Plan for the Review of Safety Analysis Reports for Nuclear Power Plants: LWR Edition*, NUREG-0800, Aug. 2016. [Online]. Available: https://www.nrc.gov/docs/ML1601/ML16019A316.pdf

[23] Nuclear Regulatory Commission, “Section 19.0: Probabilistic Risk Assessment and Severe Accident Evaluation for New Reactors,” in *Standard Review Plan for the Review of Safety Analysis Reports for Nuclear Power Plants: LWR Edition*, NUREG-0800, Dec. 2015. [Online]. Available: https://www.nrc.gov/docs/ML1508/ML15089A068.pdf

[24] Nuclear Regulatory Commission, “Regulatory Guide 1.152: Criteria for Use of Computers in Safety Systems of Nuclear Power Plants,” Jan. 2006. [Online]. Available: https://www.nrc.gov/docs/ML0530/ML053070150.pdf

[25] Canadian Nuclear Safety Commission, *Design of Reactor Facilities: Nuclear Power Plants*, REGDOC-2.5.2, Version 2, Apr. 2023. [Online]. Available: https://publications.gc.ca/collections/collection_2023/ccsn-cnsc/CC172-107-2023-eng.pdf