Assessing the Usefulness of Assurance Cases: Experience with the Large Hadron Collider

Torin Viger[1,2], Jeff Joyce[1], Simon Diemert[1], Claudio Menghi[3], Marsha Chechik[2], Jan Uythoven[4], Markus Zerlauth[4], and Lukas Felsberger[4]

[1]Critical Systems Labs, Inc.
[2]University of Toronto
[3]University of Bergamo and McMaster University
[4]European Organization for Nuclear Research (CERN)

***Abstract*—Assurance cases are structured arguments designed to show that a system is sufficiently reliable to function properly in its operational environment. They are mandated by safety standards and are largely used in industry to support risk management for systems; however, assurance cases often contain proprietary information and are not publicly available. Therefore, the benefits of assurance case development are usually not rigorously documented, measured or assessed.**

In this paper, we empirically evaluate the effectiveness of using assurance cases to show that a system is reliable using a case study over the CERN Large Hadron Collider (LHC) Machine Protection System (MPS). We used open-source documentation to create an assurance case over the MPS and used the Eliminative Argumentation methodology for its development.The development involved four authors with considerable experience in assurance case development, three of whom work for Critical System Labs, a small enterprise specializing in assurance cases. Our findings show that (a) the cost and time required to develop our assurance case is negligible compared to the effort needed to develop the system, and (b) eliminative argumentation helped identify defeaters (i.e., doubts in the system's reliability) that were not detailed in the documentation used for creation of the assurance case.

Keywords: Assurance Cases, Eliminative Argumentation, CERN, Safety Assurance, Large Hadron Collider

I. Introduction

Assurance cases (ACs) are arguments intended to show that a system will reliably function as expected in its operational environment. ACs play an important role in systems engineering as they connect technical evidence about a system to high-level claims that a wide range of stakeholders can understand, enabling engineers and reviewers to assess whether proper risk mitigations are in place. They are used in many domains (e.g., automotive, rail, and control [1]), mandated by safety standards (e.g., ISO 15026-2 [2], ISO 26262 [3], and EN 51026 [4]), and are often represented graphically (e.g., [5], [6], [7]).

Despite the interest of the research (e.g., [5], [6], [8]) and industrial (e.g., [9]) communities in ACs, the benefits of the usage of AC are seldom empirically assessed or publicly shared. On one side, the research community is usually interested in the development of new notations for AC representation (e.g., [5], [6], [8]) and automated reasoning tools (e.g., [10], [11], [12]) that are typically evaluated on showcase examples which differ significantly from those developed in the industry. On the other hand, practitioners extensively utilize ACs in their work. However, due to strict nondisclosure policies in many companies, there is limited knowledge sharing regarding practical assurance strategies and real-life industrial examples [13]. We are aware of only a handful of attempts (e.g., [1], [9], [14]) aiming to assess the benefits of using ACs in practice, and not aware of any works that evaluate the benefits of AC development over a real, publicly available AC case study.

This paper presents an empirical evaluation of the effectiveness of using ACs to show the reliability of a system. We assess the costs and benefits associated with AC development through a case study over the CERN Large Hadron Collider (LHC) Machine Protection System (MPS) [15]. The LHC is a particle accelerator and collider built by the European Organization for Nuclear Research (CERN) [16]. The LHC is a cyber-physical system combining hardware and software components [17], [18], [19]. We selected the LHC since it is a sizeable industrial case study from the nuclear domain, and we could interact with CERN engineers to empirically assess the results of our study. We relied on open-source documentation for AC creation and used *eliminative argumentation (EA)* [20] as a graphical notation. EA explicitly supports modeling defeaters, i.e., reasons to doubt AC claims. We used EA since it is a well-known methodology for AC development; it is supported by existing tools [21] and considered by Critical Systems Labs (CSL) [22] in similar works [1].

CSL is a small-medium Canadian enterprise that assesses and manages complex software safety and security risks. We developed our AC using Socrates [21], an industrial collaborative tool for AC development. Development took approximately three months and involved three engineers from CSL and one Ph.D. student with four years of experience in the assurance case domain. Our AC is a significant example comprising 506 nodes [23]. This AC is of medium size, according to industrial experience. Our AC is publicly available [24]. Unlike our previous work [23] which reports on our practical experience of creating this argument and reflects on the support provided by the features of Socrates, this paper uses this case study to assess the cost and usefulness of assurance case development.

We collected metrics and reflected on the AC creation process to answer two research questions: *What is the effort needed to develop an assurance case for a complex system* (**RQ1**)? and *How useful is the creation of an assurance case* (**RQ2**)? In terms of effort, the time (91.9 days) and estimated cost required to develop an AC for the LHC MPS are significantly lower than system development (10 years and ≈ 4.4 billion USD for construction of the LHC, of which $\approx 5\%$ was estimated to be spent on the MPS ≈ 200 million USD). To analyze usefulness, we interacted with CERN experts to understand the impact of the defeaters that our AC creation process identified but that were not detailed in the documentation available to us. CERN experts confirmed all the identified defeaters and added only a handful of new defeaters that were not included in the argument. Therefore, we conclude that EA shows high precision and recall for identifying valid defeaters.

The paper is structured as follows. Section II presents the LHC and the MPS component. Section III provides relevant background information on ACs and EA. Section IV presents the methodology used for the AC development and to evaluate each research question. Section V describes the AC for the MPS. Section VI presents our evaluation results. Section VIII discusses threats to validity. Section IX discusses related work. Section X concludes by summarizing key results and describing plans for future work.

II. The Large Hadron Collider

The *Large Hadron Collider (LHC)* is a particle accelerator and collider constructed by the European Organization for Nuclear Research (CERN). Its purpose is to test theories and explore unanswered questions in particle physics by observing collisions between highly accelerated particles. Construction of the LHC took approximately 10 years [25], [26] with material costs of approximately 4.6 billion SFr (≈ 4.4 billion USD [25]). We selected the LHC for our case study as it is a large, complex system with extensive publicly available documentation, and because CERN engineers were able to help evaluate our research questions.

The LHC accelerates particles within two 27-kilometer-long rings to nearly the speed of light in opposite directions (see Figure 1). In each ring, particle beams travel in clusters separated by particle-free gaps. The beams are bent and focused around the rings by over 10000 magnets. Collision experiments are performed by diverting the trajectories of these beams so that they collide, intersect at four collision points where the resulting phenomena are detected and analyzed by various large-scale particle detectors.

Accelerated particle beams circulating in the LHC have extremely high energy and potential destructive force. Even a single proton beam within the LHC has the power of an aircraft carrier moving at 12 knots, and if their trajectories become unstable, they pose a significant risk of damage to the system. Further, a substantial amount of energy is stored in the electrical circuits used to power the LHC magnets, and an uncontrolled release of even a small portion of this energy could result in damage to the LHC. Thus, the machine must be sufficiently protected from the damages described above.

Fig. 1. The 27 km LHC tunnel, housing the LHC accelerator, here showing the superconducting magnets containing the two beam pipes. [27].

The *Machine Protection System (MPS)* is a collection of inter-dependent components designed to prevent potential damage during LHC operations. The MPS proactively monitors all conditions that could potentially lead to damage, and protects the system by issuing a beam dump (i.e., safely extracting all particles from each ring of the LHC) before hazardous conditions are reached. Each critical component of the MPS has redundancy so that, if a failure occurs, backups of the malfunctioning MPS component will be in place to extract the beam before damage is caused.

The MPS is designed to protect the LHC from two main hazardous scenarios: beam loss and magnet quenches. The argument presented in this paper focuses on beam loss. A *beam loss* occurs when accelerated particles become unstable in their trajectory around the LHC. There are several factors that may cause beam losses, such as collisions between proton beams and residual gas molecules in the LHC ring's vacuum chamber, magnets used to bend and focus the beam around the LHC being out of tolerance, and failure to extract the beam from the one of the two LHC rings during a beam dump. This may result in a loss of containment, or particle collisions with the LHC itself. As these particles have very high energy, beam loss can cause significant damage to the LHC if it exceeds acceptable levels.

The MPS is responsible for detecting beam loss and performing beam dumps before potentially damaging conditions are reached. A *beam permit signal* is used by components of the MPS to communicate whether conditions are appropriate to continue operating the LHC: if the beam permit signal is present, the LHC may continue operating; otherwise, a beam dump is required. For this study, a simplified MPS is considered to consist of four main components: the Beam Loss Monitoring System (BLMS), the Beam Interlock System (BIS), the Beam Dumping System (BDS) and the Safe Machine Parameters (SMP).

The Beam Loss Monitoring System (BLMS) is responsible for monitoring the LHC to measure the beam loss in all portions of the ring. The BLMS consists of approximately 4000 monitors distributed around the two rings, each of which is monitoring a specific region of the LHC. Monitors are more densely distributed in critical regions of the LHC, such as around the critical components required to perform a beam dump. When non-nominal beam losses are detected, the BLMS signals the BIS to initiate a beam dump by withdrawing the beam permit. There is triple redundancy and error detection in the optical transmission to the BIS, and redundancy in other areas of the machine protection system. The MPS is intended to extract the beams within $400\mu s$ of the occurrence of a failure state to avoid potential damage to accelerator components. To satisfy this requirement, the BLMS is designed to detect and communicate beam losses to the BIS within $80\mu s$.

The Beam Dumping System (BDS) is responsible for extracting the beams from the LHC rings without damaging the system. It consists of a large graphite block designed to absorb extracted beams, dilution magnets that spread out particle clusters to reduce the energy density when they impact the sink, pulsed kicker magnets and continuously powered septa magnets to divert the circulating beams from the main LHC ring towards the sink, and moveable absorbers that protect the machine in the case of errors during a dump. For a beam dump to occur in a loss-free way, the BDS is engaged during an *abort gap*, i.e., a particle-free gap of $3\mu s$ in the ring.

If the abort gap is not particle-free or synchronisation with the abort gap is lost, the BDS will engage[1] anyway and perform an asynchronous dump. Asynchronous dumps can be dangerous as any particles that pass by the kicker magnets while they are only partially engaged will not be diverted to the proper extraction trajectory. Absorbers are placed to protect the LHC in asynchronous dumps by covering the possible trajectories that particles could be sent on if they pass by kicker magnets that are not fully engaged. The extraction time for a worst-case scenario beam dump is $178\mu s$ since it may take up to $89\mu s$ for an abort gap to synchronize with a withdrawn beam permit, and another $89\mu s$ for all the particles to be extracted from the beam.

The Beam Interlock System (BIS) determines whether the BDS should initiate a beam dump depending on the values assumed by a set of permit signals. The BIS processes these signals and sends a continuous signal to the BDS depending on the values received by the so called User systems. The BLMS is one of these User system (in total there are about 200 connections to the LHC BIS). The BIS sends a beam permit with value `true` to the BDS if it receives a beam permit with value `true` from all subsystems; otherwise, it sets the beam permit to the value `false`. It may take between $20\mu s$ and $120\mu s$ for the BIS to receive, process, and redirect a beam permit signal. Note that the BIS connects to all systems that may cause damage to the LHC. If any of these systems enter an unsafe state, the BIS will trigger a beam dump before the BLMS detects a problem. The BLMS is an additional protection measure on top of the BIS.

The *Safe Machine Parameters (SMP)* computes the values of a set of parameters from the operational conditions of the LHC and the super proton synchrotron via the two safe machine parameter controllers: one for the LHC and one for the super proton synchrotron. Once the SMPs are derived, they are communicated either via a broadcast protocol through the general machine timing channel, or via direct serial cable communications. The SMP system is used to ensure that only a low-intensity beam is injected into an LHC without a beam and that certain inputs of the BIS can be masked with safe beam intensities. It also distributes critical parameters to many systems. An example is the beam energy which is used by the BLMS to calculate the thresholds for requesting a beam dump.

[1]There are rare cases where a malfunction may cause particles to de-bunch and travel around the ring with a more uniform distribution.

III. Assurance Cases

This section provides relevant background information on ACs using a fragment of an AC from the LHC MPS.

Eliminative Argumentation (EA) [8] is a graphical notation for AC development that extends the Goal Structuring Notation (GSN) [5]. We selected EA for developing the LHC MPS AC from other alternatives (GSN, CAE, SACM [28], [29], [30]) for several reasons. First, EA enables engineers to document and reason about doubts in their argument using explicit *defeater* nodes, which emphasizes the importance of using doubt to drive critical thinking in AC development. Previous industrial projects using EA in practice have shown that it is easy to learn and that it helps facilitate critical review of uncertainties in ACs [1]. Second, though other AC development frameworks like SACM also support reasoning about doubts in an AC [31], the members of our project team were already familiar with EA, which is an important consideration in an industrially focused project. Finally, the *Socrates - Assurance Case Editor* AC tool provides robust support for EA and was made available for this project [21].

Figure 2 presents a fragment of an EA for the LHC Machine Protection System. The EA has nodes of different types:

- *Claim nodes* express affirmative statements asserting that a system satisfies one or more properties. For example, the node C0081 in Figure 2 is a claim asserting that the system's Target Dump External (TDE) dump block (i.e., a large graphite sink) will safely absorb beams from the LHC.
- *Defeater nodes* express doubts about the validity of an assurance argument. Defeaters are unique to EA, whereas the other AC node types presented in this section are also included in other notations such as GSN. A defeater can be decomposed into nodes showing how it has been mitigated, or it may be left as residual risk that threatens the argument's validity. For example, the defeater D0091 in Figure 2 asserts that claim C0081 will not hold if the TDE block absorbs a beam with high energy density that causes it to exceed its maximum safe temperature threshold. This defeater is decomposed into an argument showing that the hazardous scenario has been sufficiently mitigated.
- *Strategy nodes* express reasoning steps used to decompose a claim into more refined subclaims. For example, node S0364 in Figure 2 decomposes defeater D0091 into subclaims related to the heat load capacity and cooling of the TDE block.
- *Context nodes* are used to provide background information or missing details that may be necessary to understand the argument. For example, context X0725 in Figure 2 provides information on the maximum heat loads for each type of core in the TDE block.
- *Inference rule nodes* are attached to strategy nodes, and are used to explain the rationale for why a strategy's child

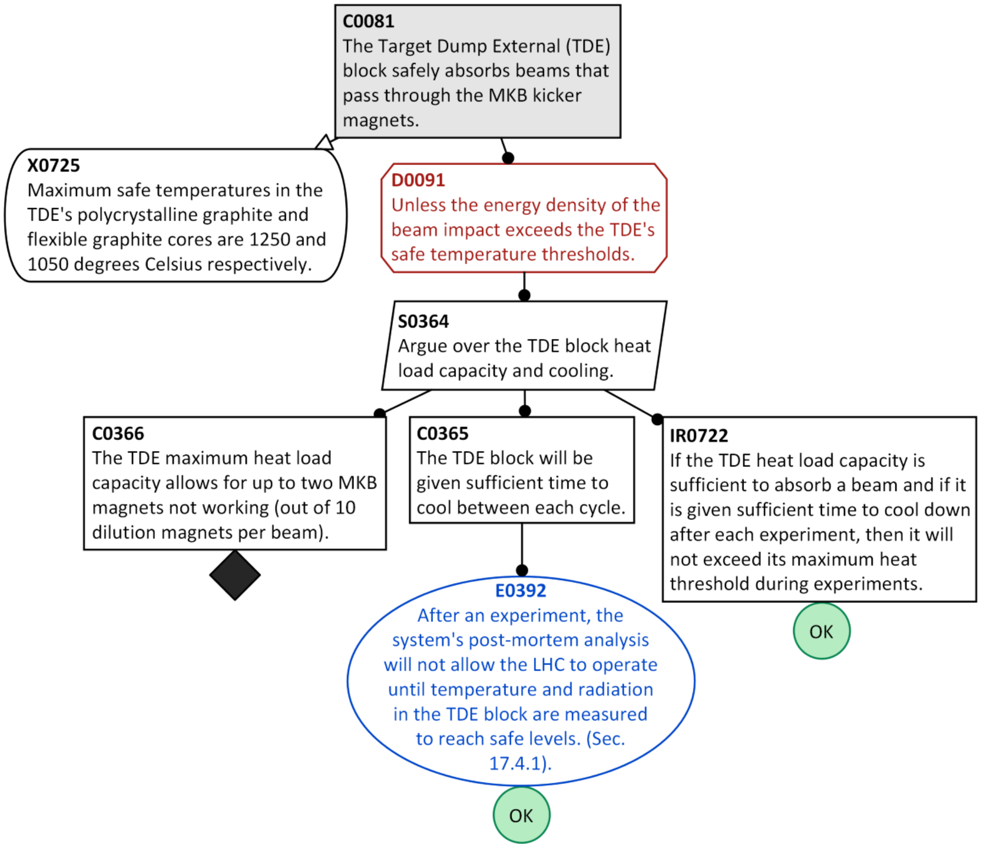

Fig. 2. AC fragment for the LHC Machine Protection System.

claims are sufficient to show that the parent claim holds. Inference rules may also be referred to as *justification* nodes (e.g., in GSN). For example, inference rule IR0722 in Figure 2 argues that if the TDE block's maximum heat load is sufficient to absorb a beam from the LHC and if it is given time to cool down each time it absorbs a beam, then it will never exceed its safe temperature threshold.

- *Assumption nodes* may be used to list conditions related to the system or its operational environment that are assumed to be true in the argument.
- *Evidence nodes* are used to support claims by directly connecting them to supporting evidence or documentation showing that the claim holds. For example, node E0392 in Figure 2 supports node C0365 by referencing a protocol in the MPS's post-mortem analysis in which it will never allow an experiment to commence when the TDE block is at a potentially unsafe temperature.
- *Residual risk nodes* are the residual uncertainties that cannot be completely eliminated by the argument, and thus they remain as potential sources of risk or uncertainty. These nodes may require further investigation or risk management strategies to mitigate their potential impact.
- *Undeveloped nodes* are aspects of the system that are not fully addressed or developed within the argument. These undeveloped nodes may require further investigation or analysis to fully understand their implications for the problem at hand. They represent areas of potential uncertainty or risk that may require further attention or consideration.

IV. Methodology

Engineers typically develop an AC following a precise methodology and development process. In this section, we describe the methodology we used to create the AC for the LHC MPS and empirically assess its benefits.

Our methodology (see Figure 3) follows two phases: *assurance case design* ($\boxed{1}$) and *feedback collection* ($\boxed{2}$). These correspond to our research questions: **RQ1** and **RQ2** from Section I.

Assurance Case Design ($\boxed{1}$). Four engineers designed the AC for the LHC. Three of them were industry experts working at CSL. The team of CSL engineers has a combined experience in AC production of over 25 years. The other is a Ph.D. student at the University of Toronto with four years of research experience in AC development. From 2009 to 2012, CSL performed a series of technical audits for CERN covering particular aspects of the MPS. Knowledge gained from earlier work assisted the effort to develop this AC. Separate from these technical audits, CSL also collaborated with CERN and Cambridge University researchers on the formal verification of a critical component of the MPS [32].

The AC design proceeded as follows:

$\boxed{1.1}$ *Collection of Material*. The AC developers conducted a literature review of public documentation for the LHC MPS and identified eight relevant papers and four technical reports – see Table I. These included engineering specifications for

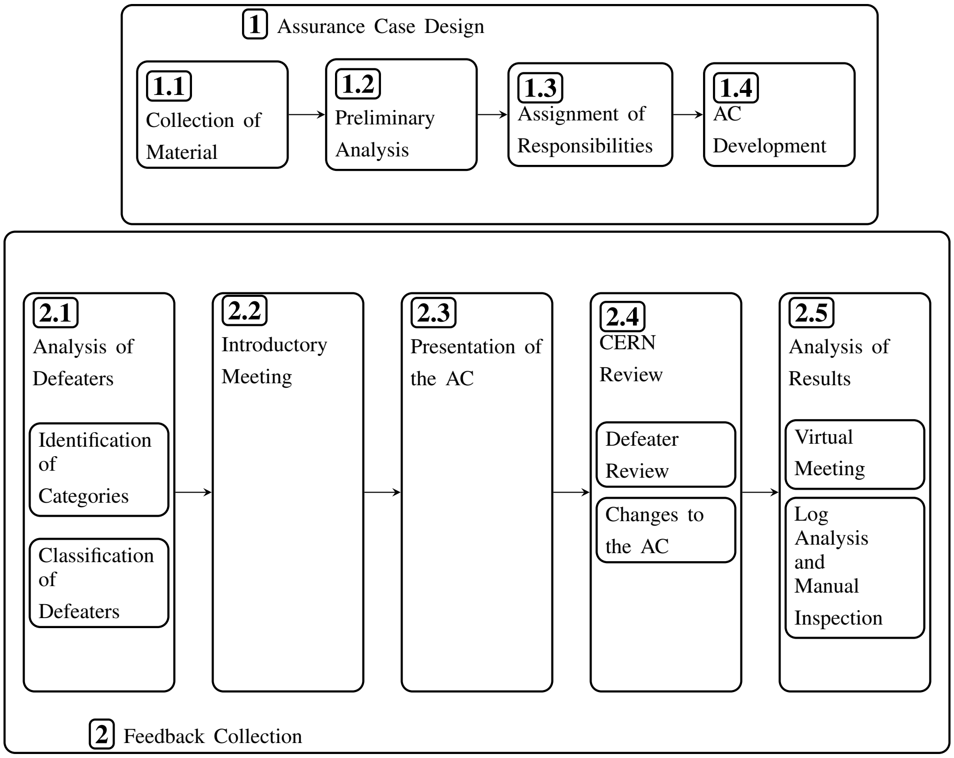

Fig. 3. Methodology for creation and analysis of the MPS AC.

TABLE I
DOCUMENTS CONSIDERED FOR THE CREATION OF THE LHC AC.

Ref	Description
[33]	Operational report for the BIS
[34]	Description of the BLMS
[35]	Documentation of the LHC beam and power interlock systems
[36]	Technical overview of the BIS
[37]	Instruments and methods for measuring beam parameters
[38]	Statistics related to operation of the BDS
[39]	Upgraded BDS configuration and behaviour of beam dumps
[40]	Ph.D. dissertation describing the LHC MPS and its components

the system captured from various CERN internal documents and reviewed/discussed in project reports (i.e., [33], [34]), scientific papers (i.e., [35], [36], [37], [38], [39]) and a Ph.D. dissertation [40].

1.2 *Preliminary Analysis of the Material*. The AC developers studied the MPS documentation with the objective to better understand the MPS and its subsystems in order to determine how AC development tasks should be distributed among the team. Collection and analysis of this material took a combined period of two weeks.

1.3 *Assignment of Responsibilities*. An online session was performed to plan the AC development and define the tasks assigned to each member of the AC development team. Each member was responsible for developing a branch of the argument for one of the four main subsystems (i.e., the BLMS, BIS, BDS, and SMP) of the MPS (see Section II).

1.4 *AC Development*. The AC design was performed using the collaborative web AC development platform Socrates [21] and took approximately seven weeks. Development was primarily done in parallel, with additional collaborative work sessions to review the argument and identify connections and interdependencies between its branches. These sessions lasted for around two hours and occurred twice weekly. Interdependencies between branches were mainly determined by considering defeaters in each argument branch and analyzing whether any other MPS subsystem performed a function that mitigated them.

The main argument creation phase was deemed complete once all branches of the argument were sufficiently decomposed so that they could be directly linked to evidence from relevant CERN documents. The process left some residual defeaters where supporting evidence could not be identified from the publicly available documentation. The argument was reviewed internally by five additional engineers for consistency and quality for two weeks. We discuss the results of the AC design and provide the answer to **RQ1** in Section VI-A.

Feedback Collection (2). The evaluation of the AC proceeded as follows:

2.1 *Analysis of Defeaters*. We performed an internal review to analyze and classify the defeaters. The internal review had the following steps:

1) *Identification of Defeater Categories*. We identified a set of categories that classify how the doubts expressed by the defeater nodes were mitigated by the documents we analyzed. The categories were initially defined by two authors and reviewed by other members of the team. The categories capture the degree to which the defeaters and their corresponding mitigations were addressed by the

TABLE II
CLASSIFICATION OF DEFEATERS.

Category	Description
RESIDUAL RISKS	The risk captured by the doubt of the defeater is not mitigated.
NOT RELEVANT	The doubt expressed by the defeater does not represent a significant risk for the system.
NOT EXPLORED	The hazard scenario is not explored in the documentation.
SOME UNDERSTANDING	The documents address the risk from the defeater without explicitly detailing it.
UNDERSTOOD	The documents detail the defeater. However, its mitigations are not simple or obvious.
WELL UNDERSTOOD	The documents precisely detail the defeater and the corresponding mitigations.

TABLE III
NUMBER OF NODES OF EACH TYPE IN THE LHC AC.

Node Type	Number of Nodes
Claims	146
Evidence	70
Strategies	32
Inference Rules	29
Context	26
Assumptions	1
Defeaters	105
Residual	9
Undeveloped	15
Complete	73
Total	506

publicly available documentation. Table II presents the categories we used to classify the defeaters. For example, the category NOT EXPLORED refers to defeaters for which corresponding hazard scenarios are not explored in CERN documents we analyzed.

2) *Classification of the Defeaters*. We associated each defeater with one of the categories defined in Table II after reviewing the CERN documentation. This process was conducted internally by project members, with suitable peer review, before being verified by CERN experts in $\boxed{2.4}$. This involved reviewing each defeater identified in the AC, analyzing the open source documentation to asses the extent that the defeater was explicitly addressed, and determining whether sufficient evidence was available to demonstrate that the scenario was satisfactorily mitigated. For example, the defeater "*Unless a pre-defined energy value for the BLMS is incorrect.*" was classified as WELL UNDERSTOOD after finding thorough documentation and evidence of the correctness of BLMS energy values.

$\boxed{2.2}$ *Introductory Meeting*. We provided a high-level presentation of the goal of our empirical study and outlined the goal of our evaluation to CERN experts.

$\boxed{2.3}$ *Presentation of the AC*. We presented the AC in detail to CERN experts to give them a general understanding of the argument we built. We then gave CERN experts access to the Socrates platform so that they could edit the AC themselves. We also shared with them the categorized list of defeaters.

$\boxed{2.4}$ *CERN Review*. Three senior CERN experts reviewed and validated the argument against existing assessments and verified the evidence for identified claims. CERN engineers provided feedback in two different ways: by reviewing the categorized list of defeaters and by directly editing the argument.

$\boxed{2.5}$ *Analysis of Results*. We held a virtual meeting with CERN engineers and collected their feedback on the categorized list of defeaters, especially those classified as NOT EXPLORED, since they capture hazard scenarios not explored in the documentation. To analyze the activity performed by CERN engineers in editing the argument, we collected and manually inspected the AC changes they made.

In the following sections, we first present the AC produced during *Assurance Case Design* ($\boxed{1}$), and then summarize the results from the *Feedback Collection* ($\boxed{2}$). Finally, during the *Feedback Analysis* review meeting, the project proposed the identification of *key performance indicators* (KPIs) from the AC. Specifically, it was proposed to use performance metrics within the AC to identify *leading* and *lagging* KPIs for the MPS. We then continued to identify subsequent KPIs for the MPS subsystems in the AC, noting areas of key performance metrics, residual or undeveloped nodes where monitoring of the system could aid mitigation of possible residual risks. Finally, the identified KPIs were shared with CERN experts for review. CERN suggested some minor typographical changes and noted that these KPIs corresponded to metrics already tracked by the post-mortem system (i.e., the system responsible for analyzing LHC data after an experiment completes).

V. THE LHC ASSURANCE CASE

The AC for the LHC created during the *Assurance Case Design* ($\boxed{1}$) has 506 nodes. Table III gives the distribution of the number of nodes of each type in the argument. Of these nodes, 105 are defeaters representing sources of doubt in the system. While most defeaters are mitigated by evidence, nine are left as residual risks within the AC.

Figure 4 presents an overview of the high-level structure of the argument. The top-level claim C0001 asserts that "The LHC Machine Protection System (MPS) protects against damage from potential beam losses" and is recursively decomposed into sub-claims, evidence, and other EA nodes. Specifically, the claim C0001 is decomposed using a strategy that splits it into four subclaims, one for each of the subsystems (i.e., BLMS, BDS, BIS, and SMP - see Section II).

Each subclaim describes how the corresponding subsystem protects against damage from potential beam losses. For example, Figure 5 presents a fragment of the AC argument associated with a subclaim for the BIS subsystem. Claim C0030 argues that "The BIS will transmit loss of the beam permit to the BDS in less than 100 microseconds". The strategy S0654 decomposes claim C0030 into four branches based on the foreseeable failure modes that could block, delay or otherwise interfere with the transmission of a beam dump request to the BDS. These failure modes are recorded explicitly by the defeater nodes D0031, D0036, D0438, and D0512. For example, the defeater D0031 argues that the BIS will transmit the loss of the beam permit to the BDS in less than 100 microseconds "Unless the beam permit loop is

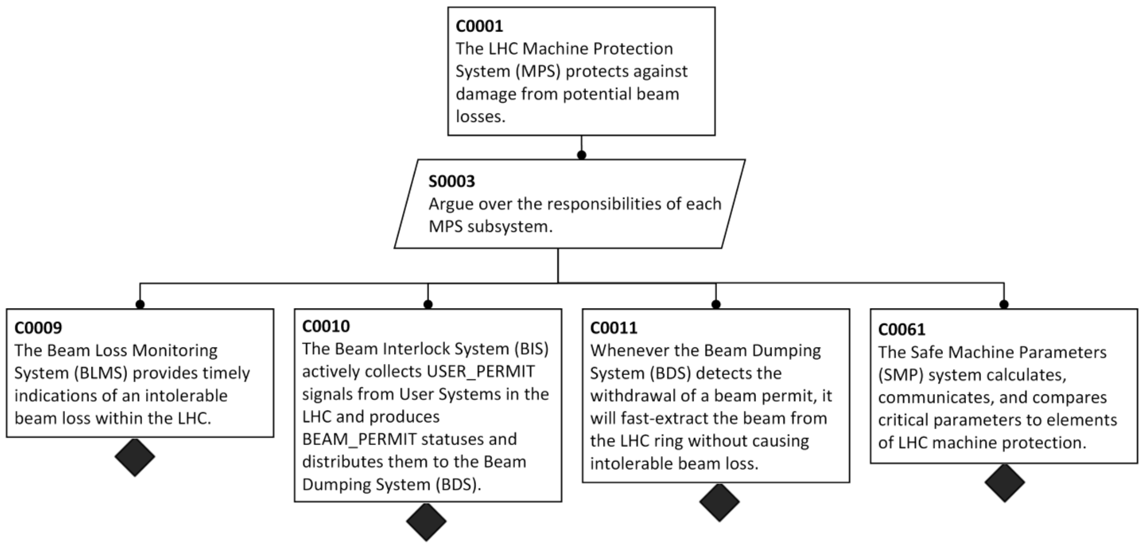

Fig. 4. Overview of the high-level structure of the argument for the LHC.

damaged in a way that interferes with the transmission of the loss of the beam permit".

EA expands defeater nodes into subclaims that terminate with evidence nodes and describe how the risks associated with the doubts introduced by the defeaters nodes are mitigated. For example, to mitigate the risks introduced by defeater D0031, the usage of redundant beam permit loops and the fail-safe design of the mechanism responsible for transmitting beam permits is considered. The evidence node E0534 of the AC asserts that "in the event of one or all transmission lines being damaged, the beam permit loop will have no 10MHz signal or noise and subsequently results in the request for a beam dump", which shows how the design of the beam permit loop mitigates risks from potential damages to the system's transmission lines.

VI. Evaluation

As discussed in Section IV, after designing the AC (*Assurance Case Design* — $\boxed{1}$), we empirically evaluated our results in collaboration with CERN experts (*Feedback Collection* — $\boxed{2}$). This section presents the answers to our research questions: Section VI-A assesses the difficulty of developing an AC for the system (**RQ1**), and Section VI-B evaluates the usefulness of the AC (**RQ2**). Finally, Section VII discusses lessons learned during the project and Section VIII presents threats to validity of our results.

A. AC Development Effort - RQ1

Our AC consists of 506 nodes, which corresponds to a medium-sized artifact according to CSL engineers. Considering a recent paper that reports on the application of EA to seven different software-intensive systems [1], our AC is larger than six of the seven ACs; the size of the remaining assurance case (513 nodes) is comparable to ours.

Developing the AC required 2543 changes (additions, modifications, and removals of nodes) and took 91.9 workdays. Table IV shows the total number of workdays spent developing

TABLE IV
Total number of workdays spent developing the MPS AC by each engineer.

Engineer Level	Activities	Days Booked
Junior A (full time)	AC creation	28.0
Junior B (full time)	AC creation	28.0
Senior (full time)	AC creation	31.4
Senior (part time)	Review, verification, validation	4.5
Total		91.9

the AC, with rows representing the time spent and activities performed by engineers at different levels. This metric includes time spent studying the MPS and its documentation. Therefore, the development time could be reduced if the AC were developed by engineers already familiar with the system.

Compared to the development time of the LHC (approximately 10 years [25], [26]), this AC development time is negligible. The AC building cost is also negligible when compared to the investment required to build the LHC MPS itself ($\approx$ 200 million USD).

RQ1 - Development Effort

The time (91.9 days) required to develop a medium-size AC for the LHC MPS is significantly smaller than the one needed for the system development (10 years). The cost estimated for developing the AC is also negligible compared to the cost of building the LHC MPS ($\approx$ 200 million USD).

B. Identifying Risk Scenarios - RQ2

To assess the usefulness of developing an AC for the MPS, we evaluate (a) whether the AC development enabled us to identify defeaters that were not explicitly detailed in the publicly available documentation and (b) the precision and recall for the identification of the defeaters of the MPS.

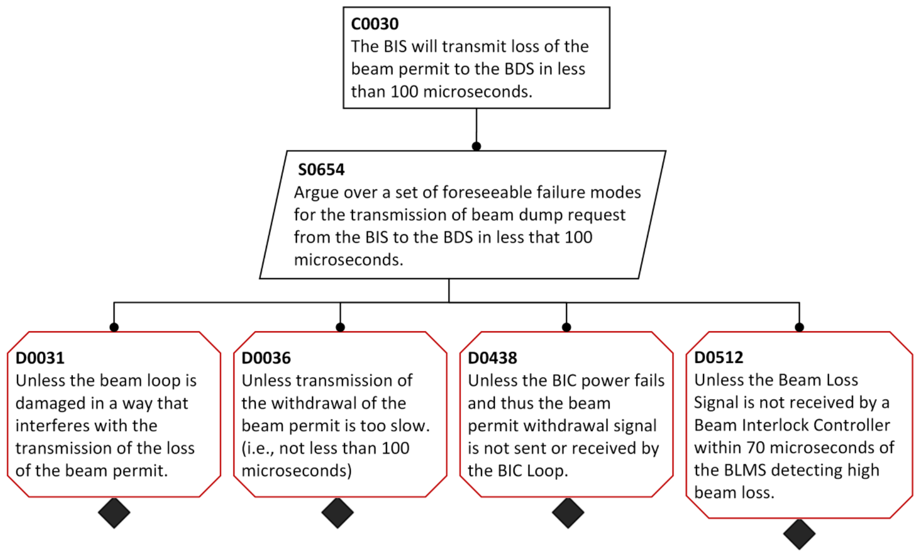

Fig. 5. Fragment of the AC that refers to the Beam Interlock System (BIS).

Table V (column *Analysis of Defeaters*) reports the number of defeaters that were classified in each of the categories from Table II during the *Analysis of Defeaters* phase ($\boxed{2.1}$) detailed in Section IV. Based on our initial classification, 24, 50, and 13 defeaters were classified as SOME UNDERSTANDING, UNDERSTOOD, and WELL UNDERSTOOD, respectively. Among the remaining 18 defeaters, nine were classified as RESIDUAL UNIDENTIFIED RISKS, three as NOT RELEVANT, and six as NOT EXPLORED. These defeaters were analyzed during the *Defeater Review* phase $\boxed{2.4}$. Table V (column *Defeater Review*) reports the number of defeaters that belongs to each category after *Defeater Review* phase. It also illustrates within brackets how this number is computed starting from the number of defeaters present in that category after the *Analysis of Defeaters*. For example, in the UNDERSTOOD row, 57 corresponds to $50 - 10 + 17$ which indicates that 10 and 17 defeaters were respectively removed and added to the 50 defeaters from the *Analysis of Defeaters*.

TABLE V
CATEGORIZATION AND NUMBER OF DEFEATERS.

Category	Analysis of Defeaters $\boxed{2.1}$	Defeater Review $\boxed{2.4}$
RESIDUAL RISKS	**9**	**7** $(9 - 2 + 0)$
NOT EXPLORED	**6**	**3** $(6 - 3 + 0)$
SOME UNDERSTANDING	**24**	**14** $(24 - 10 + 0)$
UNDERSTOOD	**50**	**57** $(50 - 10 + 17)$
WELL UNDERSTOOD	**13**	**23** $(13 - 0 + 10)$
NOT RELEVANT	**3**	**1** $(3 - 2 + 0)$
Total	**105**	**105** $(105 - 27 + 27)$

1) CERN Assurance Case Review: Recall that during the *Defeater Review* phase $\boxed{2.4}$ described in Section IV, CERN experts directly reviewed and edited the safety argument. Their review raised a total of 20 technical and editorial comments in the AC, which provided extended descriptions and clarifying details related to the design and functionality of the MPS. As an example, CERN clarified that if the BIS has no power, the beam permit loop signal should have no signal or noise, which will be interpreted as dump request by the BDS (claim C0442). Reviewing these comments resulted in the following changes:

(a) minor modifications of the AC claims to more accurately reflect the design and functionality of the MPS,
(b) creation of two additional evidence nodes,
(c) revision of three nodes from claims to defeaters,
(d) creation of one new defeater for a previously unexplored branch of the BLMS argument focused on the potential for beam energy to not be processed correctly by the BLMS,
(e) one defeater being marked as UNDEVELOPED in the AC, which focused on a potential scenario involving a sudden loss in power to the BDS and available documentation,
(f) addition of two context nodes to the argument to expand on information provided by CERN experts on the operation of the BLMS.

2) Defeater Review: In total, 27 defeaters were reclassified following the Defeater Review by CERN experts. The impact on each category of defeaters was as follows:

- RESIDUAL UNIDENTIFIED RISKS. CERN experts confirmed seven of the defeaters in this category but noted that the remaining two were mitigated by additional publicly available information about the MPS which they described during a review meeting.
  Resulting Changes: Two defeaters were removed from the RESIDUAL UNIDENTIFIED RISKS category and moved to the UNDERSTOOD category.
- NOT RELEVANT. After consulting the additional references identified by CERN experts (research papers and supporting documentation from publicly accessible CERN resources), information was found related to the mitigation of two defeaters initially classified as NOT RELEVANT category. The remaining NOT RELEVANT defeater was confirmed by CERN experts as not a relevant

risk to the LHC MPS.

Resulting Changes: Two defeaters were removed from the Not Relevant category and moved to the Understood category.

- Not Explored. CERN experts explained the measures used to mitigate the risk associated with three defeaters that were classified into this category. They then confirmed the relevance of the remaining three defeaters as well as absence of the mitigation measures for them in the publicly available documents we considered.

  Resulting Changes: Three defeaters were removed from the Not Explored category and moved to the Understood category.
- Some Understanding. CERN experts explained how the measures reported in the documents we analyzed mitigated the risk associated with ten defeaters initially categorized under Some Understanding. They then confirmed the relevance of the remaining fourteen defeaters in this category, for which we could not find thorough mitigation measures in the documentation we analyzed.

  Resulting Changes: Ten defeaters were removed from the Some Understanding category and moved to the Understood category.
- Understood and Well Understood. CERN experts confirmed the relevance of all defeaters in these categories and provided additional information which enabled us to improve the AC and expand on the mitigation measures for ten of the defeaters initially classified as Understood.

  Resulting Changes: Ten defeaters were removed from the Understood category and moved to the Well Understood category.

The results from Table V (column **Defeater Review**) show that only one defeater was classified as Not Relevant after the *Defeater Review*. Among the remaining 104 defeaters, $\approx$ 90% (95 = 14 + 57 + 23) were classified as Some Understanding, Understood, or Well Understood and were known by CERN experts. This result is expected since the AC development was based on an existing operating system and publicly accessible online documentation containing information reported by CERN experts. Of the remaining $\approx 10\%$, seven were classified as Residual Unidentified Risks and three were classified as Not Explored. These defeaters were confirmed to be relevant by CERN experts, and not explicitly detailed in the publicly available documentation. This result is significant: it shows the usefulness of AC development in identifying real defeaters that could impact the reliability of a system, and the level of precision of the approach. Therefore, we conclude that development of an AC using EA is useful to accurately identify doubts in a system.

To calculate the precision and recall of our identification of defeaters, we defined True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) as follows:

- TPs are defeaters that we classified as relevant during the *Analysis of Defeaters* phase ($\boxed{2.1}$) (i.e., those in the Residual Unidentified Risks, Not Explored, Some Understanding, Understood and Well Understood categories) which were confirmed to be relevant by CERN experts during the *Defeater Review* phase ($\boxed{2.4}$). All defeaters in these categories were confirmed to be relevant, therefore TP $= 102$ $(9+6+24+50+13)$.
- TNs are defeaters which we classified as Not Relevant that were confirmed to be Not Relevant by CERN experts after the *Defeater Review*. We have TN $=$ 1, as only one of the three defeaters identified as Not Relevant was confirmed by CERN to be not relevant.
- FPs are defeaters which we categorized as relevant defeaters (i.e., in any category except Not Relevant), but that CERN identified as Not Relevant. We had no false positives, therefore FP $= 0$.
- FNs are nodes which we did not categorize as relevant defeaters, but which CERN identified to be relevant. We have two FNs from the *Defeater Review* phase ($\boxed{2.4}$), as two defeaters we categorized as Not Relevant were found to be relevant by CERN experts. Additionally, as noted in Section VI-B1, CERN changed three nodes from claims to defeaters and added an entirely new defeater during their review of the AC itself. Therefore, FN $= 6$ $(2+3+1)$.

The precision of our defeater identification process is $^{TP}/_{(TP+FP)} = {}^{102}/_{102+0} = 1$.
Our recall is $^{TP}/_{(TP+FN)} = {}^{102}/_{(102+6)} = 0.94$.

> **RQ2 - Usefulness**
>
> The answer to RQ2 is that AC development identified 10 defeaters that were not detailed in the publicly available documentation we considered. These defeaters were confirmed by CERN experts. The precision of the manual development process we used to identify defeaters is 1, the recall is 0.94.

VII. Discussion and Lessons Learned

This section discusses lessons learned during development of the CERN LHC MPS AC.

Retrospective vs. Concurrent Assurance Case Development. Development of this AC was a retrospective effort intended to capture why the original developers of the CERN LHC MPS were confident that the system, as designed, would provide adequate protection in the event of a dangerous beam loss. While many details of their reasoning can be found in various technical reports, publications, presentations, and other artifacts, this effort aimed to show how these details connect to higher-level safety properties and to capture the reasoning structure. A *retrospective AC*, such as the one described in this paper, is valuable for preserving important knowledge as key personnel retire or move away to different responsibilities. However, it is possible (and generally the standard practice) to develop an AC concurrently with the original development of a system, service, or product. Concurrent development of the AC is more likely to capture details while they are fresh in the minds of the developer, and concurrent development can

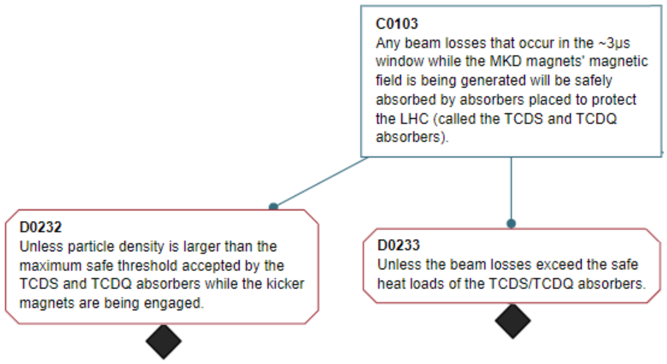

Fig. 6. A claim over the BDS beam absorbers and its associated defeaters.

inform design decisions and evidence collection of the system itself [41].

Identifying Interdependencies Using EA. The four main branches of the MPS AC were developed in parallel by four different engineers. Each branch primarily focuses on a different subsystem of the MPS, though there are also interactions and interdependencies between subsystems which are essential to the argument. As an example, consider the AC fragment from the BDS argument branch shown in Fig. 6. The claim *C0103* asserts that beam losses will be safely absorbed by BDS components. This claim can be supported with evidence showing that the heat capacity of the BDS absorbers is sufficient to absorb a beam with regular particle density; however, the BDS is not responsible for maintaining the beam's particle density. Claim *C103* may not hold if a beam has abnormal characteristics, and therefore, this claim's correctness depends on factors beyond the control of the BDS. This doubt was explicitly incorporated into the argument as a defeater (*D0232*).

To identify and address interdependencies, developers met weekly to collaboratively review each branch of the AC. When a branch was found to depend on reasoning or evidence elsewhere in the AC, the developers added an explicit *cross-reference* to link relevant external nodes to the branch. Collaborative review was essential because, as is common when developing large industrial ACs, technical knowledge about the AC and about each MPS subsystem was distributed across multiple team members. This process led to the identification of six explicit *cross-references* between different branches of the AC. We found that all six of the identified cross-references fell underneath defeater nodes, with four directly mitigating defeater nodes and two supporting a claim one level below a defeater. This reflects our experience collaboratively reviewing the AC, where defeater nodes—particularly residual defeaters—formed a central focus of team discussions. Defeaters were often left residual when they could not be sufficiently mitigated by a single subsystem, requiring input from team members with more extensive knowledge of other MPS subsystems to mitigate.

For example, defeater *D0232* in Fig. 6 was ultimately mitigated by referencing a portion of the BLMS argument branch which explains how the LHC beam is monitored to ensure that it maintains regular particle density. Residual defeaters not only served to raise important questions about each individual subsystem, but also helped uncover these broader interdependencies across the AC. This highlights the value of defeaters in guiding collaborative review and providing useful insights during AC development.

Common Terminology and Patterns. Development of the CERN AC demonstrated the importance of having common, consistent terminology when creating a large AC with multiple developers. In early drafts of the CERN AC, multiple terms were used to represent similar concepts (e.g., "signal", "communicate", and "provide" all being used to describe the transfer of information from one subsystem to another). This introduced challenges in the AC review process, as reviewers were unclear whether these variations in terminology were intentional and meaningful. To mitigate this challenge, we defined a consistent set of common terms and refactored the AC using these terms to decrease inconsistency and ambiguity. The lessons learned from this process led to the subsequent development of a new *glossary* feature in Socrates, which allows key terms to be defined in a glossary in the AC editor itself and referenced throughout the AC. Maintaining terminological consistency is a challenge that we have observed to be applicable in AC development across multiple domains, and may benefit from automated Large Language Model (LLM) support, i.e., using LLMs to identify discrepancies in terminology across large ACs and propose ways to resolve them [42].

Pattern Templates have been proposed to support AC development—patterns enable commonly used argument structures to be reused in different contexts [43], and many AC development tools and notations enable pattern creation and instantiation (e.g., the SACMN extension of SACM [44]). The structure of the LHC MPS AC was designed to reflect the structure of the MPS itself: top-level system properties were decomposed into properties over its subsystems and further decomposed into properties of individual components; however, we did not explicitly use any established patterns in the development of our AC. The CERN LHC is a one-of-a-kind system, and thus the availability and applicability of AC patterns was lesser than it may have been if developing an AC in other domains with more common systems (e.g., automotive vehicles). One area in which patterns could have been leveraged to enhance the AC is in referencing evidence. Most evidence throughout the argument consisted of references to publicly available CERN documentation. Developing reusable patterns for citing different types of documented information (e.g., design specifications, calculations, hazard analysis, etc.) could have improved clarity and consistency throughout the AC. Recently, developers of Socrates added a new feature that enables patterns to be created and instantiated within the AC editor itself [21].

Defeater Categories: Defeaters can be classified into different categories based on the type of node they challenge: Undercutting defeaters challenge inference rules, undermining defeaters challenge evidence, and rebutting defeaters challenge claims. Recent work by Gohar et al. [45] proposes an extended taxonomy for defeater classification which divides defeaters into seven categories based on the nature of the doubt they express (e.g., structural defeaters, adversarial defeaters etc.). While our work did not analyze the distribution of defeaters across different taxonomies, assessing the extent to which defeaters of different types were identified and/or mitigated gives an interesting direction for future work.

The goal of this paper was to empirically evaluate the effectiveness of using an AC to show the reliability of a system. Comparing AC development against alternative approaches to reliability analysis was out of scope for this work; however, our case study can provide a benchmark for such comparisons in future work. We also note that ACs are not intended to replace traditional techniques for safety and reliability assessment such as FMEA [46] or FTA [47], but rather to integrate their results within a structured argument. These methods are often used in conjunction with AC development to identify relevant hazards, structure the argument, and collect supporting evidence.

VIII. Threats to Validity

Threats to validity. The methodology used to collect the feedback from CERN, i.e., the defeater review and the changes to the AC via Socrates, is an internal threat to validity of our results. To mitigate this, we used two methods to collect feedback: the discussion of the defeater review and the analysis of the changes performed via Socrates on the AC. Another internal threat to validity is team composition and experience of team members. To mitigate this, we created a team composed of members with a mix of experience, including industry and academia.

The analysis of a single case study threatens the external validity of our results: the conclusions of our empirical investigation may differ for different case studies and systems. However, the fact that the MPS is a large safety-critical system and the involvement of experts in AC development from CSL mitigates this threat: a large safety-critical system is likely to share problems that are also encountered in other safety-critical systems, and the presence of CSL engineers ensured that the creation of the AC was grounded on previous experience. Usage of public documentation for the AC development and analysis is another external threat to validity since for other systems (still under development), this documentation may not be available, or might be incomplete. Therefore, when systems are not as mature as the one we analyzed, we expect precision and recall to be lower. Further, an AC is normally created during the system design and hence evolves over time, this would also likely affect the precision and recall.

Finally, the metrics used to measure the usefulness of the defeaters produced by the manual development process (TP, FP, TN, and FN) threaten our results' construction validity since they influence how well they represent or reflect a concept that is not directly measurable.

IX. Related Work

There is significant research and industry interest in approaches that support AC development, including new notations [28], [8], [5], methodologies [9], [48], [49], [50], argument templates [51], [43], [52], domain-specific techniques [53], [14], and tools for formal reasoning over ACs [54], [11], [55], [10]. However, these techniques are often not assessed or only assessed over small showcase examples. For example, the work introducing EA [8] demonstrated its usefulness on three artificial examples with approximately 30 nodes. As Habli et al. note [56], there is a lack of systematic evaluation of AC methods, including when, how, and why they provide benefits in practice. They emphasize the importance of grounding claims about the value of ACs in empirical evidence; we take a step toward addressing this gap in our work by empirically analyzing the benefits of an AC developed using EA. We only identified a handful of works that analyze the implications and effectiveness of AC development techniques in practice. We summarize these works below.

Diemert and Joyce [1] discussed their experiences and lessons learned from using EA to create ACs for seven different industrial systems, including four automotive (149, 257, 484 and 513 nodes), two rail (40 and 95 nodes) and one industrial control (14 nodes) AC. The authors report that EA increases confidence in ACs and helps with independent safety assessment, though the complete ACs and information on their development processes are not made fully publicly available. In addition, the lessons learned are presented informally, whereas our work gives a systematic empirical analysis of our AC development process.

Recent work by Borg et al. [57] reports on the development of an AC for an automotive pedestrian emergency braking system with an ML-based perception component. Their work demonstrates how the *Assurance of Machine Learning for use in Autonomous Systems (AMLAS)* framework [58] can be used in conjunction with the ISO 21448 SOTIF standard [59] to develop ACs in practice, and highlights lessons learned during the development process such as the challenges and benefits of using a simulator to create datasets. The ALMAS framework has subsequently been applied by Sivakumar et al. [60] to create an AC for a reinforcement learning-enabled Component of a Quanser Autonomous Vehicle. While neither of these works empirically evaluate the effectiveness of their resulting ACs or their cost to develop, they each provide a complete worked example that can serve as a foundation for future research on ML safety assurance. The resultant ACs from each work are also made open-source.

Sujan et al. [61] reviewed AC practices in six UK industries (automotive, civil aviation, defense, nuclear, petrochemical, and railway). Their analysis compares safety requirements and regulations from the healthcare domain and concludes that ACs may lead to more structured healthcare safety management practices; however, the authors note that further research studies are required to provide empirical evidence of the contribution of ACs to safety management.

Graydon and Holloway [62] reviewed twelve candidate proposals (in fifteen papers) for assessing confidence in ACs. Their goal was to assess the capabilities of the proposed techniques for quantifying confidence in assurance arguments. The authors searched for counterexamples to detect techniques that can produce implausible results. Where possible, the authors prioritized counterexamples that are variants of the original examples. For three out of twelve techniques, the authors reported some counterexamples showing that the technique outputs are untrustworthy.

Nair et al. [63] used evidential reasoning [64] to measure and aggregate confidence in ACs. Evidential reasoning requires safety analysts to attach confidence levels to evidence nodes to denote the evidence's trustworthiness and aggregates these values to derive a quantified confidence measurement in the AC. The authors evaluated their framework through a survey involving 21 participants with over two years of experience in safety assurance. However, the authors did not assess their framework in any case study.

Cyra et al. [65] proposed visual assessment to analyze an argument that relies on the Dempster-Shafer theory of evidence [66]. Dempster–Shafer's theory of evidence requires associating each evidence node with a value within the interval $[0, 1]$. Strategies are linked to functions that enable computing the confidence of the different claims from the confidence of evidence nodes. The authors analyzed whether the functions associated with the different strategies are plausible. This analysis was conducted via an experiment involving 31 students from the Master's degree in information technologies. The results show that the accuracy of the aggregation rules is similar to the consistency from the answers of the participants.

Unlike these works, this paper empirically analyzed and assessed the benefits of an AC developed using EA on a significant industrial example. We made our AC and results publicly available.

Large Language Models and Safety Assurance: Recent work has begun exploring the potential for Large Language Models (LLMs) [42] to augment and/or automate various tasks related to AC development. Oluwafemi et al. [67], [68] introduced the SmartGSN tool which uses LLMs to support multiple stages of AC development, including instantiation of GSN ACs from pre-defined patterns, pattern detection within ACs and generation of graphical ACs from text. Sivakumar et al. [69] introduced a framework for AC generation by prompting LLMs to create AC fragments based on manually written system descriptions, a top-level argument structure and a description of available evidence. They evaluated their framework by assessing the model's ability to reproduce three "ground-truth" argument fragments from real ACs, and found the model had "moderate" success in this task.

While LLMs have the potential to partially automate the creation of ACs, this use-case also introduces risks that the LLM will *hallucinate*, i.e., invent facts or evidence which give confidence in a false conclusion. In our parallel work [70], we argue that the risks associated with LLM hallucinations are largely mitigated when models are used to support *defeater identification* rather than argument creation. LLMs have the potential to help AC developers identify novel defeaters that they may have otherwise overlooked, e.g., due to blind spots or confirmation bias. Our recent work [71] provides a framework for systematically structuring and evaluating LLM-generated defeaters. We evaluated this framework over six different ACs developed by Critical Systems Labs (including the CERN AC presented in this paper). Our findings show that (i) LLMs can generate defeaters which are informative, relevant and useful, (ii) practitioners found LLM-generated defeaters to be helpful in supporting defeater identification, and (iii) AC development using LLM defeater generation led to a broader range of doubts being identified by engineers in practice. Other groups have also explored the potential for LLMs to support defeater generation. Their findings support this proposed use-case by showing that LLMs are effective at generating similar defeaters to those identified by humans during AC development [72], [73].

This paper analyzed the effort needed to develop an AC for a complex system (RQ1) and the usefulness of this AC (RQ2). Our focus was on assessing how the AC helped improve system safety by identifying doubts that were not explicitly addressed in publicly available documentation. Recent work [74] proposed seven potential benefits of AC development which can be used as metrics to assess usefulness, such as how ACs facilitate communication between stakeholders, enhance documentation of system properties, and enable rigorous evaluation of evidence and assumptions. Assessing these additional benefits is out of scope: We plan to consider them in future work.

This paper shares the same nuclear case study and assurance case as our previous works [23] [75]. Unlike our previous works, which reflect on the support provided by the features of Socrates to create an AC and the use of EA to identify Key Performance Indicators, this work defines a precise and detailed methodology that is used to empirically assess the benefits of AC development on a large and representative industrial case study (Section IV). Following this methodology, we collected empirical data, analyzed this data, computed the recall precision of AC development on our case study, and presented our findings (Section VI). Finally, we presented the threats to the validity of our results and discussed the role of eliminative argumentation in identifying KPIs (Section VIII).

X. CONCLUSION

In this paper, we empirically evaluated the effort required to develop an AC for a large representative safety-critical system, and assessed its usefulness in showing that a system is reliable. Our results show that the cost and time required to develop this AC are negligible compared to the system cost, the AC helped identify risk scenarios not explicitly considered in the documentation considered for the AC development, and the manual development was effective in identifying defeaters. Based on our knowledge, this is the first study that empirically assesses the usefulness of the AC and involved two teams of experts from the industries, one helping in the construction of the AC (experts from CSL) and one for its evaluation (experts from CERN). Therefore, our results will be relevant to both researchers, who need industrial case studies to assess their research solutions, and practitioners, who can rely on empirical results confirming the usefulness of AC development.

Our AC is publicly available, as is all of its supporting documentation. Based on our knowledge, this is the only publicly available AC that includes more than 100 nodes, developed in collaboration with safety experts with an extensive experience in safety analysis and revised by domain experts.

The development of a large and representative exemplar AC is part of our long-term vision of supporting AC developers by using AC development information as data [76]. We believe that by monitoring the AC development activities and treating ACs as data, we can learn suggestions to help safety engineers improve their AC. For this reason, the activity of the AC developers was monitored, and their activities during the creation of the AC were logged. We plan to treat these AC development activities as data and to propose techniques that can learn from these data and provide suggestions that can improve the design of the AC. For example, recommendations may help identify safety interdependencies between the components for the MPS overlooked during the AC design.

Acknowledgment

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) [funding reference numbers RGPIN-2022-04622, DGECR-2022-0040, RGPIN-2015-06366].

The work of Claudio Menghi was partly supported by the European Union - Next Generation EU, "Sustainable Mobility Center (Centro Nazionale per la Mobilità Sostenibile - CNMS)", M4C2 - Investment 1.4, Project Code CN_00000023, by projects SERICS (PE00000014) under the NRRP MUR program, and GLACIATION (101070141).

References

[1] Diemert Simon, Joyce Jeff. Eliminative Argumentation for Arguing System Safety - A Practitioner's Experience in *Proceedings of International Systems Conference*:1–7IEEE 2020.

[2] ISO/IEC JTC 1/SC 7 Software and systems engineering . Systems and software engineering — Systems and software assurance — Part 2: Assurance case https://www.iso.org/standard/52926.html 2011.

[3] Palin Rob, Ward David, Habli Ibrahim, Rivett Roger. ISO 26262 Safety Cases: Compliance and Assurance in *International Conference on System Safety*IET 2011.

[4] BS EN 50126-1:2017 2011.

[5] GSN Working Group . GSN Community Standard Version 2 http://www.goalstructuringnotation.info/ 2011.

[6] Boomfield Robin, Bishop Peter. Safety and Assurance Cases: Past, Present and Possible Future – an Adelard Perspective in *Safety-Critical Systems Symposium*Springer 2010.

[7] Rushby John. The Interpretation and Evaluation of Assurance Cases *Comp. Science Laboratory, SRI International, Tech. Rep. SRI-CSL-15-01.* 2015.

[8] Goodenough John, Weinstock Charles, Klein Ari. Eliminative Argumentation: A Basis for Arguing Confidence in System Properties Tech. Rep. CMU/SEI-2015-TR-005Software Engineering Institute, Carnegie Mellon UniversityPittsburgh, PA 2015.

[9] Mohamad Mazen, Åström Alexander, Askerdal Örjan, Borg Jörgen, Scandariato Riccardo. Security Assurance Cases for Road Vehicles: An Industry Perspective in *International Conference on Availability, Reliability and Security*ACM 2020.

[10] Viger Torin, Murphy Logan, Di Sandro Alessio, Menghi Claudio, Shahin Ramy, Chechik Marsha. The ForeMoSt Approach To Building Valid Model-Based Safety Arguments *Software and Systems Modeling.* 2022:1–22.

[11] Fung Nick L. S., Kokaly Sahar, Di Sandro Alessio, Chechik Marsha. Assurance Case Property Checking with MMINT-A and OCL in *Recent Trends and Advances in Model Based Systems Engineering*:351–360Springer 2022.

[12] Viger Torin, Murphy Logan, Di Sandro Alessio, Shahin Ramy, Chechik Marsha. A Lean Approach to Building Valid Model-Based Safety Arguments in *ACM/IEEE International Conference on Model Driven Engineering Languages and Systems* 2021.

[13] Cheng Jinghui, Goodrum Micayla, Metoyer Ronald, Cleland-Huang Jane. How Do Practitioners Perceive Assurance Cases in Safety-Critical Software Systems? in *International Workshop on Cooperative and Human Aspects of Software Engineering*:57–60ACM 2018.

[14] Jaradat Omar, Sljivo Irfan, Habli Ibrahim, Hawkins Richard. Challenges of Safety Assurance for Industry 4.0 in *European Dependable Computing Conference*IEEE 2017.

[15] Schmidt Rüdiger, Assmann R, Carlier E, et al. Protection of the CERN Large Hadron Collider *New Journal of Physics.* 2006;8:290.

[16] The Large Hadron Collider https://home.cern/science/accelerators/large-hadron-collider 2022 [Online].

[17] Andersson Riccard, Adli Erik, Bargalló Enric, Nordt Annika. Machine Protection Systems and their Impact on Beam Availability and Accelerator Reliability in *International Particle Accelerator Conference*:MOPTY044 2015.

[18] Holzer Eva Barbara, Dehning Bernd, Effnger Ewald, et al. Beam Loss Monitoring for LHC Machine Protection *Physics Procedia.* 2012;37:2055-2062. International Conference on Technology and Instrumentation in Particle Physics.

[19] Dehning B. LHC Machine Protection in *Beam Instrumentation Workshop* 2008.

[20] Goodenough John B, Weinstock Charles B, Klein Ari Z. Eliminative induction: A basis for arguing system confidence in *International Conference on Software Engineering*:1161–1164IEEE 2013.

[21] Socrates Assurance Case Editor https://criticalsystemslabs.com/socrates-assurance/ 2025 [Online].

[22] Critical Systems Labs https://www.criticalsystemslabs.com/ 2022 [Online].

[23] Millet Laure, Diemert Simon, Rees Chris, et al. Assurance Case Arguments in the Large: the CERN LHC Machine Protection System in *Computer Safety, Reliability, and Security (SAFECOMP)*Springer 2023.

[24] Rees Chris, Viger Torin, Delgado Mateo, et al. CERN LHC MPS Assurance Case https://cds.cern.ch/record/2854725 2023.

[25] Large Hadron Collider https://en.wikipedia.org/wiki/Large_Hadron_Collider 2022 [Online].

[26] Highfield Roger. Large Hadron Collider: Thirteen ways to change the world *The Daily Telegraph.* 2008:10–10.

[27] CERN Website Images https://home.web.cern.ch/about 2023 [Online].

[28] Gan S, Ryan JA. Claims, arguments, evidence *International Nuclear Information System (INIS).* 2019;52.

[29] Nemouchi Yakoub, Foster Simon, Gleirscher Mario, Kelly Tim. Isabelle/SACM: Computer-Assisted Assurance Cases with Integrated Formal Methods in *Integrated Formal Methods*:379–398Springer 2019.

[30] Selviandro Nungki. Assurance Case Pattern using SACM Notation in *2021 9th International Conference on Information and Communication Technology (ICoICT)*:494–499 2021.

[31] Shahandashti Kimya Khakzad, Belle Alvine B, Lethbridge Timothy C, Odu Oluwafemi, Sivakumar Mithila. A PRISMA-driven systematic mapping study on system assurance weakeners *Information and Software Technology.* 2024:107526.

[32] Ghafari Naghmeh, Kumar Ramana, Joyce Jeff, Dehning Bernd, Zamantzas Christos. Formal verification of real-time data processing of the LHC beam loss monitoring system: a case study in *Formal Methods for Industrial Critical Systems*:212–227Springer 2011.

[33] Puccio B, Romera Ramirez I, Todd B, Kwiatkowski M, Castañeda Serra A. The CERN beam interlock system: principle and operational experience tech. rep.European Organization for Nuclear Research (CERN) 2010.

[34] Gilardoni SS, Effinger E, Gil-Flores J, Wienands U, Aumon S. Beam loss monitors comparison at the CERN proton synchrotron tech. rep.European Organization for Nuclear Research (CERN) 2011.

[35] Bordry Frederick, Schmidt R, Mess KH, Rodríguez-Mateos F, Puccio B, Denz R. Machine protection for the LHC: Architecture of the beam and powering interlock systems tech. rep.European Laboratory for Particle Physics, European Organization for Nuclear Research (CERN) 2001.

[36] Puccio B, Schmidt R, Wenninger J, others . Beam interlocking strategy between the LHC and its injector in *International Conference on Accelerator and Large Experimental Physics Control Systems*:10–14 2005.

[37] Gasior M, Jones R, Lefevre T, Schmickler H, Wittenburg Kay. Introduction to beam instrumentation and diagnostics *arXiv preprint arXiv:1601.04907.* 2016.

[38] Carlier Etienne, Bracco C, Wiesner C, et al. LHC beam dumping system *Evian Workshop on LHC beam operation.* 2017:215-220.

[39] Maestre J, Torregrosa C, Kershaw K, et al. Design and behaviour of the Large Hadron Collider external beam dumps capable of receiving 539 MJ/dump *Journal of Instrumentation.* 2021;16:P11019.

[40] Wagner Sigrid Claudia. *LHC machine protection system: method for balancing machine safety and beam availability*. PhD thesisETH Zurich 2010.

[41] Hobbs Christopher, Diemert Simon, Joyce Jeff. Driving the development process from the safety case in *Safe AI Systems*Safety-Critical Systems Club 2024.

[42] Naveed Humza, Khan Asad Ullah, Qiu Shi, et al. A comprehensive overview of large language models *arXiv preprint arXiv:2307.06435.* 2023.

[43] Szczygielska Monika, Jarzebowicz Aleksander. Assurance case patterns on-line catalogue in *Advances in Dependability Engineering of Complex Systems: International Conference on Dependability and Complex Systems*:407–417Springer 2018.

[44] Selviandro Nungki. Assurance case pattern using sacm notation in *2021 9th International Conference on Information and Communication Technology (ICoICT)*:494–499IEEE 2021.

[45] Gohar Usman, Hunter Michael C, Cohen Myra B, Lutz Robyn R. A Taxonomy of Real-World Defeaters in Safety Assurance Cases *arXiv preprint arXiv:2502.00238.* 2025.

[46] McDermott Robin E, Mikulak Raymond J, Beauregard Michael R. Fmea *New York: Taylor & Francis Group.* 2009.

[47] Ericson Clifton A, Ll Clifton. Fault tree analysis in *System Safety Conference, Orlando, Florida*;1:1–9 1999.

[48] Javed Muhammad Atif, Muram Faiz Ul, Hansson Hans, Punnekkat Sasikumar, Thane Henrik. Towards Dynamic Safety Assurance for Industry 4.0 *Journal of Systems Architecture.* 2021;114:101914.

[49] Viger Torin, Salay Rick, Selim Gehan, Chechik Marsha. Just enough formality in assurance argument structures in *International Conference on Computer Safety, Reliability, and Security*:34–49Springer 2020.

[50] Inge JR. The safety case, its development and use in the United Kingdom in *Equipment Safety Assurance Symposium 2007* 2007.

[51] Chowdhury Thomas, Lin Chung-Wei, Kim BaekGyu, Lawford Mark, Shiraishi Shinichi, Wassyng Alan. Principles for Systematic Development of an Assurance Case Template from ISO 26262 in *Proceedings of 2017 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*:69-72 2017.

[52] Yamamoto Shuichiro, Matsuno Yutaka. An evaluation of argument patterns to reduce pitfalls of applying assurance case in *International Workshop on Assurance Cases for Software-Intensive Systems*:12–17IEEE 2013.

[53] Myklebust Thor, Stålhane Tor, Hanssen G. Agile safety case and DevOps for the automotive industry in *European Safety and Reliability Conference and the Probabilistic Safety Assessment and Management Conference* 2020.

[54] Arcaini Paolo, Bombarda Andrea, Bonfanti Silvia, Gargantini Angelo, Riccobene Elvinia, Scandurra Patrizia. The ASMETA approach to safety assurance of software systems in *Logic, Computation and Rigorous Methods: Essays Dedicated to Egon Börger on the Occasion of His 75th Birthday*:215–238Springer 2021.

[55] Denney Ewen, Pai Ganesh. Tool support for assurance case development *Automated Software Engineering.* 2018;25:435–499.

[56] Habli Ibrahim, Alexander Rob, Hawkins Richard David. Safety cases: An impending crisis? in *Safety-Critical Systems Symposium (SSS'21)* 2021.

[57] Socha Kasper, Borg Markus, Henriksson Jens. SMIRK: A machine learning-based pedestrian automatic emergency braking system with a complete safety case *Software Impacts.* 2022;13:100352.

[58] Hawkins Richard, Paterson Colin, Picardi Chiara, Jia Yan, Calinescu Radu, Habli Ibrahim. Guidance on the assurance of machine learning in autonomous systems (AMLAS) *arXiv preprint arXiv:2102.01564.* 2021.

[59] Schnellbach Adam, Griessnig Gerhard. Development of the ISO 21448 in *European Conference on Software Process Improvement*:585–593Springer 2019.

[60] Sivakumar Mithila, Belle Alvine B, Shan Jinjun, Odu Oluwafemi, Yuan Mingfeng. Design of the safety case of the reinforcement learning-enabled component of a quanser autonomous vehicle in *2024 IEEE 32nd International Requirements Engineering Conference Workshops (REW)*:57–67IEEE 2024.

[61] Sujan Mark A, Habli Ibrahim, Kelly Tim P, Pozzi Simone, Johnson Christopher W. Should healthcare providers do safety cases? Lessons from a cross-industry review of safety case practices *J. Safety Science.* 2016;84:181–189.

[62] Graydon Patrick J, Holloway C Michael. An investigation of proposed techniques for quantifying confidence in assurance arguments *Safety science.* 2017;92:53–65.

[63] Nair Sunil, Walkinshaw Neil, Kelly Tim, Vara Jose Luis. An evidential reasoning approach for assessing confidence in safety evidence in *International Symposium on Software Reliability Engineering*:541–552IEEE 2015.

[64] Yang Jian-Bo, Xu Dong-Ling. On the evidential reasoning algorithm for multiple attribute decision analysis under uncertainty *Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans.* 2002;32:289–304.

[65] Cyra Lukasz, Górski Janusz. Support for argument structures review and assessment *Reliability Engineering & System Safety.* 2011;96:26–37.

[66] Shafer Glenn. Dempster-shafer theory *Encyclopedia of artificial intelligence.* 1992;1:330–331.

[67] Odu Oluwafemi, Beltrán Daniel Méndez, Gutiérrez Emiliano Berrones, Belle Alvine B, Sherafat Melika. SmartGSN: a generative AI-powered online tool for the management of assurance cases *arXiv preprint arXiv:2410.16675.* 2024.

[68] Odu Oluwafemi, Belle Alvine B, Wang Song, Kpodjedo Segla, Lethbridge Timothy C, Hemmati Hadi. Automatic instantiation of assurance cases from patterns using large language models *arXiv preprint arXiv:2410.05488.* 2024.

[69] Sivakumar Mithila, Belle Alvine B, Shan Jinjun, Shahandashti Kimya Khakzad. Prompting GPT–4 to support automatic safety case generation *Expert Systems with Applications.* 2024;255:124653.

[70] Viger Torin, Murphy Logan, Diemert Simon, Menghi Claudio, Di Alessio, Chechik Marsha. Supporting assurance case development using generative ai in *SAFECOMP 2023, position paper* 2023.

[71] Viger Torin, Murphy Logan, Diemert Simon, et al. AI-Supported Eliminative Argumentation: Practical Experience Generating Defeaters to Increase Confidence in Assurance Cases in *2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE)*:284–294IEEE 2024.

[72] Gohar Usman, Hunter Michael C, Lutz Robyn R, Cohen Myra B. CoDefeater: Using LLMs To Find Defeaters in Assurance Cases *arXiv preprint arXiv:2407.13717.* 2024.

[73] Shahandashti Kimya Khakzad, Sivakumar Mithila, Mohajer Mohammad Mahdi, Belle Alvine B., Wang Song, Lethbridge Timothy C.. Evaluating the Effectiveness of GPT-4 Turbo in Creating Defeaters for Assurance Cases arXiv 2024.

[74] Rinehart David J, Knight John C, Rowanhill Jonathan. Understanding What It Means for Assurance Cases to" Work" tech. rep. 2017.

[75] Rees Chris, Casey Adam, Joyce Jeff, et al. A discussion on the use of Eliminative Argumentation (EA) to identify Key Performance Indicators (KPIs) for the CERN LHC Machine Protection System :2461-2468 2023.

[76] Menghi Claudio, Viger Torin, Di Sandro Alessio, Rees Chris, Joyce Jeff, Chechik Marsha. Assurance Case Development as Data: A Manifesto in *International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*:135-139IEEE/ACM 2023.