applied sciences

MDPI

Article

Combining Model-Based Systems Engineering and Knowledge-Centric Systems Engineering to Design Reliable Systems in Practice

Juan Manuel Morote [1], Jose Luis de la Vara [2,*], Giovanni Giachetti [3,4], Clara Ayora [2] and Luis Alonso [5]

[1] Independent Researcher, 02003 Albacete, Spain
[2] Departamento de Sistemas Informáticos, Universidad de Castilla-La Mancha, Avda. España s/n, 02071 Albacete, Spain; clara.ayora@uclm.es
[3] Facultad de Ingeniería, Universidad Andrés Bello, Antonio Varas 880, Providencia, Santiago 7591538, Chile; giovanni.giachetti@unab.cl
[4] Valencian Research Institute for Artificial Intelligence (VRAIN), Universitat Politècnica de València, Camí de Vera, s/n, 46022 Valencia, Spain
[5] The REUSE Company, C/ Margarita Salas 16, 28919 Madrid, Spain; luis.alonso@reusecompany.com
* Correspondence: joseluis.delavara@uclm.es

Abstract

The use and importance of complex software-intensive systems are growing. As they are used in a wider range of situations in which dependability must be ensured, the reliability of the systems and of their components needs to be addressed throughout their lifecycle, including at early development stages. In addition, the means used to deal with reliability need to be linked to and integrated into the overall systems engineering practices and processes. Within this context, we present an approach to design reliable systems in practice in the scope of model-based systems engineering (MBSE) and knowledge-centric systems engineering (KCSE), two systems engineering perspectives whose adoption is increasing. While MBSE relies on explicit system models, KCSE places artificial intelligence at its core to capture, formalise, and reason over system knowledge. Both perspectives are combined to model systems and analyse whether their design addresses the expected system reliability properties, leveraging knowledge representation, natural language processing, and inference mechanisms. The approach links the processes and tools of Arcadia/Capella for MBSE and of SES Engineering Studio for KCSE. A joint application process has been defined for system modelling, ontology development, structured textual requirements specification, traceability management, and model quality analysis, all of which are targeted at system reliability. For validation, the approach has been applied on eight systems that cover five different application domains, considering tens of diagrams, of knowledge elements, of reliability properties, and of analysis possibilities. Based on the validation results, we argue that the approach is a feasible means to design reliable systems. The approach is also the first one that effectively combines MBSE with Arcadia/Capella and KCSE with SES to design reliable systems in practice.

Keywords: model-based systems engineering; knowledge-centric systems engineering; reliability; system design; artificial intelligence; Arcadia; Capella; SES Engineering Studio

Academic Editors: Linda Vickovic and Maja Braović

Received: 21 January 2026
Revised: 19 February 2026
Accepted: 19 February 2026
Published: 24 February 2026

1. Introduction

Software-intensive systems pervade society nowadays. For example, cyber-physical systems, automated systems, autonomous systems, and software-intensive electronic sys-

tems and components play a major role in communications, Industry 4.0, healthcare, and transport, making our lives easier and better. The range of functions and applications of the systems is increasing, leading to a larger set of usages in which dependability must be ensured [1]. For many systems, it is not acceptable that they fail or that their failure can pose an unreasonable risk. In critical domains, dependability is addressed in accordance with industrially agreed best practices presented in engineering and assurance standards [2], e.g., DO-178C [3] in aerospace, ISO 26262 [4] in automotive, and EN 50128 [5] in railway. The quality of the systems needs to be guaranteed so that they can be deemed dependable, including their reliability, and so that they are allowed to operate.

Reliability can be defined as the ability of systems and components to perform their required functions under stated conditions for a specified period of time [6]. In essence, reliability relates to the extent to which a system behaves well and without failure. Reliability needs to be considered during the whole system lifecycle, from inception until decommission. For example, systems can be monitored during operation to analyse if their behaviour is as defined. At early development stages, reliability must be addressed when a system is envisioned, specified, and designed, i.e., when engineers make decisions about how the system will be. It is at these stages that the expected functionality and operation conditions of a system are defined, as well as how they will be satisfied. Means to characterise and analyse system reliability according to the decisions made must be used.

The means used to characterise and analyse system reliability must also be linked to and integrated into the overall systems engineering practices and processes. We focus on two perspectives currently used in industry, whose adoption is growing, and that can be used together: model-based systems engineering (MBSE) [7] and knowledge-centric systems engineering (KCSE) [8]. In MBSE, models that represent systems and their elements (e.g., in diagrams) are used to specify, analyse, realise, and manage system artefacts. MBSE is commonly applied for complex systems in a wide range of application domains, such as aerospace, automotive, defence, and railway. KCSE specialises in MBSE from the main principle that systems and software engineering processes can exploit knowledge bases about systems and their lifecycles. These knowledge bases, which are traditional artificial intelligence (AI) means [9], can correspond to ontologies that characterise system domains and consider different types of knowledge elements and of semantic aspects. The growing complexity of system architectures and the need for sound knowledge require approaches that can capture, structure, and reason over diverse engineering knowledge. KCSE meets this requirement by using AI techniques such as ontologies and automated reasoning to facilitate systematic analysis, traceability, and decision-making. These capabilities are particularly relevant in the field of reliability engineering, where implicit expert knowledge, cross-domain dependencies, and early-stage uncertainty must be managed explicitly. The ontologies can be employed for, e.g., system specification and analysis. KCSE also usually exploits other traditional AI techniques, e.g., natural language processing (NLP). The use of AI for different tasks is positioned as a key aspect for current and future systems and software engineering (see e.g., [10–13]).

We present an approach that combines MBSE and KCSE to design reliable systems in practice. The approach integrates (a) the Arcadia method and the Capella tool for MBSE [14] and (b) SES Engineering Studio [15] (hereafter referred to as SES) for KCSE. Arcadia is an MBSE method for systems, hardware, and software architectural design developed by Thales, and Capella is the tool that implements and supports Arcadia, providing methodological guidance and intuitive model edition features. SES is a tool to orchestrate the development of all kinds of systems (hardware, software, hybrid. . .), supporting interoperability between an unlimited number of existing systems engineering tools. SES integrates several (sub-)tools for specific KCSE tasks, e.g., ontology management and system artefact

quality analysis. Many companies from different application domains use and support Arcadia/Capella and SES in the industry, most often companies that develop critical systems. Nonetheless, a reliability-oriented joint application of Arcadia/Capella and SES has not been defined. Our approach has been developed in the scope of two large-scale industry-academia European projects: iRel40 (intelligent Reliability 4.0) [16] and VALU3S (Verification and Validation—V&V—of Automated Systems' Safety and Security) [17]. The projects dealt with how to increase the reliability of modern systems and applications and with how to improve the way suitability and quality of safety- and security-critical systems are confirmed, respectively. Industrial partners from the projects expressed the need for and their interest in combining Arcadia/Capella and SES.

The approach defines how Arcadia/Capella and SES can be combined for the design of reliable systems in practice. It also considers compliance with standards. The approach consists of five activities: (1) System modelling, (2) Ontology development, (3) Structured textual requirements specification, (4) Traceability management, and (5) Model quality analysis. The activities address different reliability needs and use Arcadia/Capella, SES, or both, as well as AI. For validation, the use of the approach on eight systems has been studied, covering five different application domains and tens of diagrams of knowledge elements, of reliability properties, and of analysis possibilities.

The main contribution of our work is the development and application of an approach that effectively combines MBSE with Arcadia/Capella and KCSE with SES to design reliable systems in practice. To the best of our knowledge, other researchers have not presented such results. Practitioners and researchers interested in or working with MBSE and KCSE can benefit from the approach, and they can now know more precisely how to use Arcadia/Capella and SES together for reliable-system design, as well as how this joint use has been enacted for specific systems. Beyond the contributions on tool interoperability, the approach sets principles on how to specify reliability data in Arcadia/Capella diagrams and how to use reliability-related information in different KCSE tasks, with the ultimate goal of designing reliable systems.

In addition, we extend our prior work. First, we outlined our vision to link Arcadia/Capella and SES for model-based reliability-oriented system design [18], introducing possibilities and alternatives. Later, we presented the possibilities and alternatives selected and their initial application [19]. The main extensions in this paper from these prior publications are: (a) the update of the approach according to the latest versions of Arcadia/Capella and of SES; (b) a revised, more thorough description of the activities of the approach, including major details not presented yet such as all the reliability characteristics that can be specified in Arcadia/Capella diagrams and the existing SES quality analyses that can be used, and; (c) a full validation considering different systems in depth and covering all the activities of the approach. This results in an updated, complete, and validated description of the approach, which facilitates its application and provides evidence of its effectiveness.

The next sections present the background of our work (Section 2), the approach (Section 3), its validation (Section 4), and our main conclusions (Section 5).

2. Background

The background is divided into Arcadia/Capella, SES, and related work.

2.1. Arcadia/Capella

Arcadia/Capella [14] is an Eclipse-based open-source MBSE solution. While Arcadia serves as a methodology for architecture engineering of systems, software, and hardware, Capella (Figure 1) functions as the implementation tool for Arcadia. Major companies,

including Thales, GMV, and Siemens, are among those that utilise Arcadia/Capella in their operations.

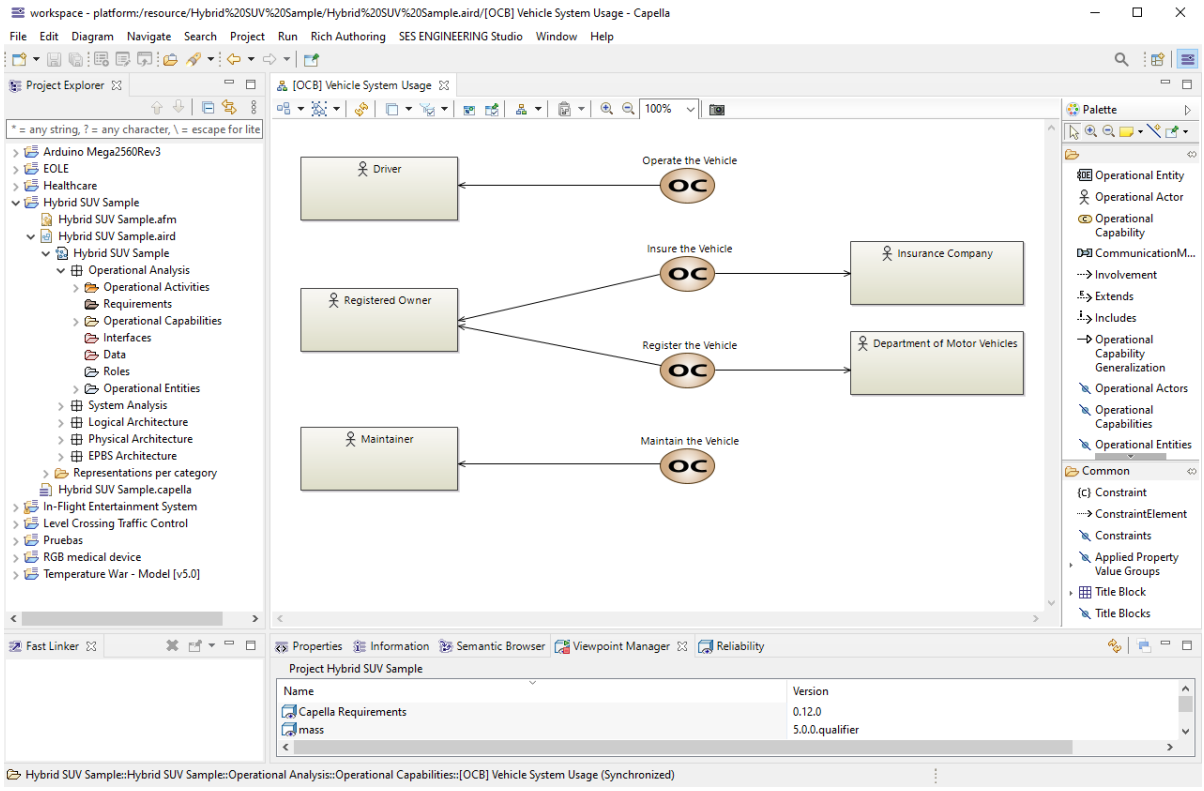

Figure 1. Capella screenshot.

Capella implements specific functionalities to follow the Arcadia method, such as the method explorer and a wide range of modelling accelerators and aids, so that systems engineers can focus on design instead of on model maintenance. Capella is a modelling tool that provides a wide range of diagram types organised into the five engineering steps defined in Arcadia, and whose use can be tailored to project- and system-specific needs and characteristics:

1. Operational Analysis captures the goals and conditions that system users need to meet in their work or mission, without consideration of any particular solution or system to accomplish these goals. The main diagrams of this step are the Operational Capabilities diagram, Operational Activity Interaction diagram, Operational Activity Breakdown diagram, and Operational Architecture diagram.
2. System Analysis specifies how the system is expected to meet the users' needs, as outlined in the prior Operational Analysis or in the form of requirements provided by a client. The main diagrams of this step are the System Functional Dataflow diagram, System Functional Breakdown diagram, System Architecture diagram, Functional Scenario diagram, and Exchange Scenario diagram.
3. Logical Architecture represents the initial design decisions of a solution's architecture. The main diagrams of this step are the Logical Functional Breakdown diagram, Logical Functional Dataflow diagram, Logical Component Breakdown diagram, and Logical Architecture diagram.
4. Physical Architecture provides a detailed specification of all the subsystems or components to be developed or acquired. It also defines the system integration, verification, and validation phases. The main diagrams of this step are the Physical Functional

Breakdown diagram, Physical Functional Dataflow diagram, Physical Component Breakdown diagram, and Physical Architecture diagram.

5. End-Product Breakdown Structure (EPBS) establishes the expectations for each system component and specifies the conditions for their integration, leading up to product V&V. The main diagrams in this step are the Configuration Items Breakdown diagram and EPBS Architecture diagram.

Capella offers extension and integration mechanisms, e.g., viewpoints to extend and enhance system specification. Expert knowledge is required to define viewpoints for specific domains, which are implemented using Capella Studio [20]. Viewpoints define extensions that are specific to the model elements managed by Capella, define additional properties to the elements, and can add a new group of commands to the palette of the diagram editor.

2.2. SES Engineering Studio

SES [15] is a commercial environment for KCSE that different large companies use in their systems and software engineering processes, e.g., Airbus, Toyota, and Ariane, among others. It is a software solution specifically created to coordinate the development of diverse system types, including hardware, software, and hybrid systems. SES facilitates seamless interaction between many existing systems engineering tools, such as tools for requirements management, MBSE, simulation, risk management, RAMS (reliability, availability, maintainability, and safety) management, and MS Office.

SES integrates various (sub-)tools to support key KCSE activities, most notably:

- RQA—Quality Studio (hereafter referred to as RQA) for system artefact quality analysis. It allows users to define, calculate, manage, and report quality for any engineering item that can be accessed through any available connection in SES.
- V&V Studio for V&V management. V&V is connected to the notion of quality assurance and management. V&V Studio merges the three concepts, managing the V&V actions necessary for quality assessment processes.
- RAT—Authoring tools (hereafter referred to as RAT) for text-based system specification. It is a tool for system analysts and engineers during the writing or model creation process.
- Traceability Studio for traceability management. It provides support to manage and specify traces between elements of system artefacts and automatically identify potentially missing traces.
- Knowledge Manager for ontology management. It supports the development and maintenance of knowledge bases from a systems engineering standpoint, enabling the storage of valuable system information in a shared system knowledge repository.

When entering SES, the user can select the different artefacts that they want to manage for KCSE, considering the different functionality provided by the (sub-)tools. Connections are established with the artefact sources, e.g., Excel spreadsheets and Capella projects. All in all, SES can be regarded as an expert, rule-based system for systems and software engineering that exploits semantic information about a domain and that uses AI techniques such as NLP and machine learning.

SES exploits ontologies that consist of five distinct parts [21]:

1. Terminology, which includes the terms used in a specific domain and their syntactic information. For instance, the word ‘car’ is a noun.
2. Conceptual Model, which focuses on semantics and relationships. For instance, the semantics of ‘car’ can be ‘system’ and it specialises ‘vehicle’.
3. Patterns, which are templates (aka boilerplates) used for structured system information specification and analysis.

4. Formalisation, which involves the semantic representation of system information according to patterns.
5. Inference Rules, which are procedures utilised to derive information, such as determining the correctness of specifications based on the structure and content of textual requirements and model elements.

Each part of an ontology uses elements of the previous ones. The elements of the different parts are employed for quality analysis, structured textual specification, and traceability management.

2.3. Related Work

Related work can be divided into general work that has dealt with, e.g., reliability aspects in MBSE or ontology-based systems and software engineering, work with Arcadia/Capella, and work with SES.

As the work on and interest in MBSE has increased, it has become easier to find general work on, e.g., model-based safety analysis [22], V&V [23], risk management [24], and traceability [25]. Pieces of work on system modelling and reliability can be found for specific languages and tools such as AADL [26], Papyrus [27], Rhapsody [28], and SysML [29], but not integrated with KCSE. It has also become easier to find publications on the use of ontologies to support different systems and software engineering activities, e.g., safety, security, and dependability risk assessment [30], requirements engineering [31], and software reliability modelling [32].

Closer to this paper, some publications have used both models and ontologies for systems and software engineering. The goals of these publications include the definition and usage of ontologies for software [33] and V&V [34] processes in general, as well as more specific purposes. For instance, Dong et al. [35] proposed an ontology and semantic rules for traceability management in MBSE, Jinzhi et al. [36] explored the concept of cognitive digital twin, and Peugeot [37] presented the GONG environment for ontology-based MBSE with UML class diagrams. Visions and initiatives have also been reported in the context of ESA [38] and NASA [39]. Nonetheless, the structure and content of the ontologies used in prior general work are different to those exploited by SES, which consist of further element types, are based on KCSE industrial needs and practices in several domains and enable a more detailed and broader system specification and analysis possibilities. In addition, to the best of our knowledge, the publications that have combined models and ontologies for systems and software engineering have not focused on the design of reliable systems in compliance with standards.

Different publications have presented work with Arcadia/Capella, addressing aspects areas such as cybersecurity risk assessment [40] and integration, verification, and validation [41], also considering safety [42], safety-security [43], and reliability aspects [44]. Some authors have addressed solutions and usages more specific than only the application of Arcadia/Capella in a given case. For instance, Brau et al. [45] proposed a solution for satellite system availability evaluation on the basis of its physical architecture, and Brunel et al. [46] defined a viewpoint for formal safety and security assessment of system architectures with Alloy. It has been mentioned that there is an interest in an ontology-driven framework for simulation model development connected with Arcadia/Capella [47], but such a solution does not appear to have been created.

The joint use of Arcadia/Capella with other engineering methods and tools has also been presented, e.g., with Cyber Architect [43], STPA (System-Theoretic Process Analysis) [48], and Valispace [49]. These pieces of work aim to provide broader, more suitable solutions that mitigate the limitations of the different individual methods and tools. This is in line with our integration with SES, which aims for Arcadia/Capella and

SES to benefit each other to design reliable systems. Regarding commercial tools, some (e.g., ATICA [50]) support model-based safety analysis with Capella, covering analyses for faults, failures, and reliability in general. As indicated below in Section 3, these kinds of tools complement our work. In essence, they can provide input about reliability data and needs to consider for, e.g., system architecture design. Although the work on and interest in Arcadia/Capella is growing both in academia and in industry, we are not aware of any publication by other authors that has reported in detail the link of Arcadia/Capella with KCSE.

In relation to work with SES, integration of SES and system modelling has been reported or referred to in the past. Historically, structured textual requirements specification with RAT and requirements quality analysis with RQA have been proposed as possible parts of MBSE processes (e.g., [51]), although not integrated as such. For instance, requirements specification on diagrams was not possible. Nonetheless, SES sub-tools have evolved during the last few years to enable full integration.

In the scope of the AMASS project (Architecture-driven, Multi-concern and Seamless Assurance and Certification of Cyber-Physical Systems) [52], we worked on different aspects of the integration of KCSE with SES and MBSE. For assurance and certification purposes, RQA was integrated with a method and tool for model-based assurance evidence management [53] and was used for quality analysis of safety cases [54]. The use of SES features for traceability and for semantic information search was also studied [55]. For artefact quality analysis [8], progress was made in relation to requirements metrics (e.g., based on the Conceptual Model part of an ontology), model metrics for Simulink, SysML, and UML diagrams, and the use of checklists for manual artefact analysis, in addition to analysis of artefact quality evolution [56]. We also worked on ontology development in the context of compliance with standards [57], but not considering the combination of MBSE and KCSE.

Attention has been paid to facilitate model reuse with SES for different system modelling languages, e.g., SysML [58]. This is based on the more effective semantics-based searches that SES can enable when compared to other MBSE tools such as Rhapsody and Papyrus. Tool interoperability [59] and trace discovery [60] have been and are a major concern for SES, including interoperability with MBSE tools and model element traceability.

As can be observed, progress has been made towards the integration of system modelling and SES, especially recently. However, in-depth work on the link of Arcadia/Capella and SES to design a reliable system has only been presented in our prior publications, which present the previous, reduced versions of our approach.

3. Approach Description

Our approach to designing reliable systems in practice combines (a) MBSE with Arcadia/Capella and (b) KCSE with SES. It mainly exploits available means in these solutions, including AI-based SES capabilities. The approach adapts how the means are used to the specific needs of reliable-system design and provides an integrated process that combines MBSE and KCSE. The approach is also aligned with how practitioners use Arcadia/Capella [61] and SES [62].

As shown in Figure 2, the approach has been divided into five activities for System modelling, Ontology development, Structured textual requirements specification, Traceability management, and Model quality analysis, all of which are targeted at system reliability. Structured textual requirements specification, Traceability management, and Model quality analysis use the results from System modelling and Ontology development. It must be noted that SES is a proprietary solution; thus, all the tool support that the approach

exploits cannot be shared via, e.g., an open-source repository, and full disclosure of SES implementation details is not possible.

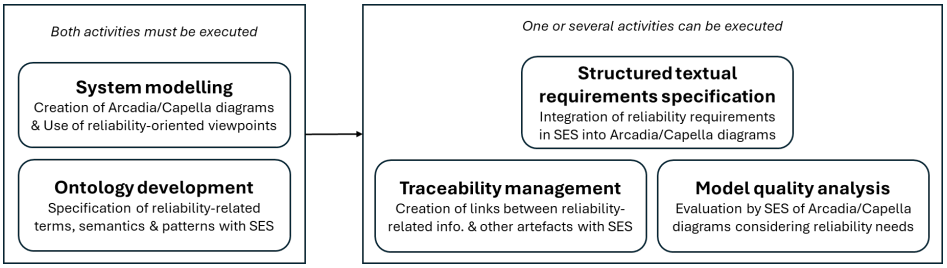

Figure 2. Overview of the approach.

The following sub-sections describe the approach activities, using an in-flight entertainment (IFE) system as a running example. IFE refers to the entertainment available to aircraft passengers during a flight, which must be reliable and safe. With the sometimes miles of wiring involved, voltage leaks and arcing can become a problem. For instance, an IFE system was implicated in the crash of Swissair Flight 111 in 1998 [63]. To address possible issues, IFE systems are typically isolated from the main systems of an aircraft. There exist public examples of IFE models [14] and of reference requirements [64].

3.1. System Modelling

The initial step of the approach involves creating Arcadia/Capella diagrams. The selection of Arcadia/Capella steps and diagram types will vary depending on the goal and extent of the modelling project. For instance, if the requirements are managed in DOORS, there might not be a need to model requirements in Arcadia/Capella diagrams. For the IFE system, a Physical Architecture diagram contains different physical components such as the passenger remote control, video server unit, and private video display unit (Figure 3). Arcadia/Capella diagrams could be part of the Software Requirements Data and the Design Description that DO-178C requires.

An important feature of our approach is the extension of Capella, allowing the specification of reliability information for diagram elements (Figure 3; Reliability Attributes below). We have created viewpoints with the Capella Studio development environment (Figure 4) that can be used for different systems. Certain diagrams may contain information regarding anticipated reliability features, while other diagrams may depict actual features. Such actual features are represented in physical architecture diagrams. The determination of reliability information will follow the specification, decomposition, and refinement steps of Arcadia, from high-level characteristics in logical diagrams to lower-level characteristics in physical diagrams.

For the development of the viewpoints, first, we studied which diagrams could be correctly extended in Capella Studio. Once this was confirmed, we examined for which elements of the diagram it made sense to add reliability information. We then chose what viewpoints would be developed and the information to be included using them. After the implementation, the behaviour of the viewpoints was tested within Capella Studio. Finally, the viewpoints were installed in Capella and activated in the corresponding diagrams.

The Arcadia/Capella diagrams and their respective elements to which reliability information can be added using the developed viewpoints are as follows:

- Logical Architecture diagram: Logical Components and Logical Actors.
- Physical Architecture diagram: Physical Components and Physical Actors.
- EPBS Architecture: Configuration Items.
- System Functional Dataflow diagram: System Functions.

- Logical Functional Dataflow diagram: Logical Functions.
- Physical Functional Dataflow diagram: Physical Functions.

Two different viewpoints have been developed: one with reliability and environment information for the Logical Architecture, Physical Architecture, and EPBS Architecture diagrams, and another for the functions in the System Functional Dataflow, Logical Functional Dataflow, and System Functional Dataflow diagrams. The first viewpoint consists of two parts. The first part contains general attributes of reliability, while the second part contains attributes on environmental aspects. The attributes have been selected from existing sources, e.g., [65–68]. The information to be added to the viewpoint can be adapted to the types of elements or systems under development. Appendix A lists the complete set of reliability attributes (e.g., Mean Time Between Failure and Failures in Time) and of environmental aspect attributes (e.g., Temperature and Humidity) considered. The second viewpoint applies to the system functions, logical functions, or physical functions of dataflow diagrams. It adds the attribute "response time" to those functions.

As an example, the IFE system requires voltage considerations for its elements [64], which can be specified in Capella using the viewpoint. The value of viewpoint attributes related to failure rates may vary depending on the assurance level of the system.

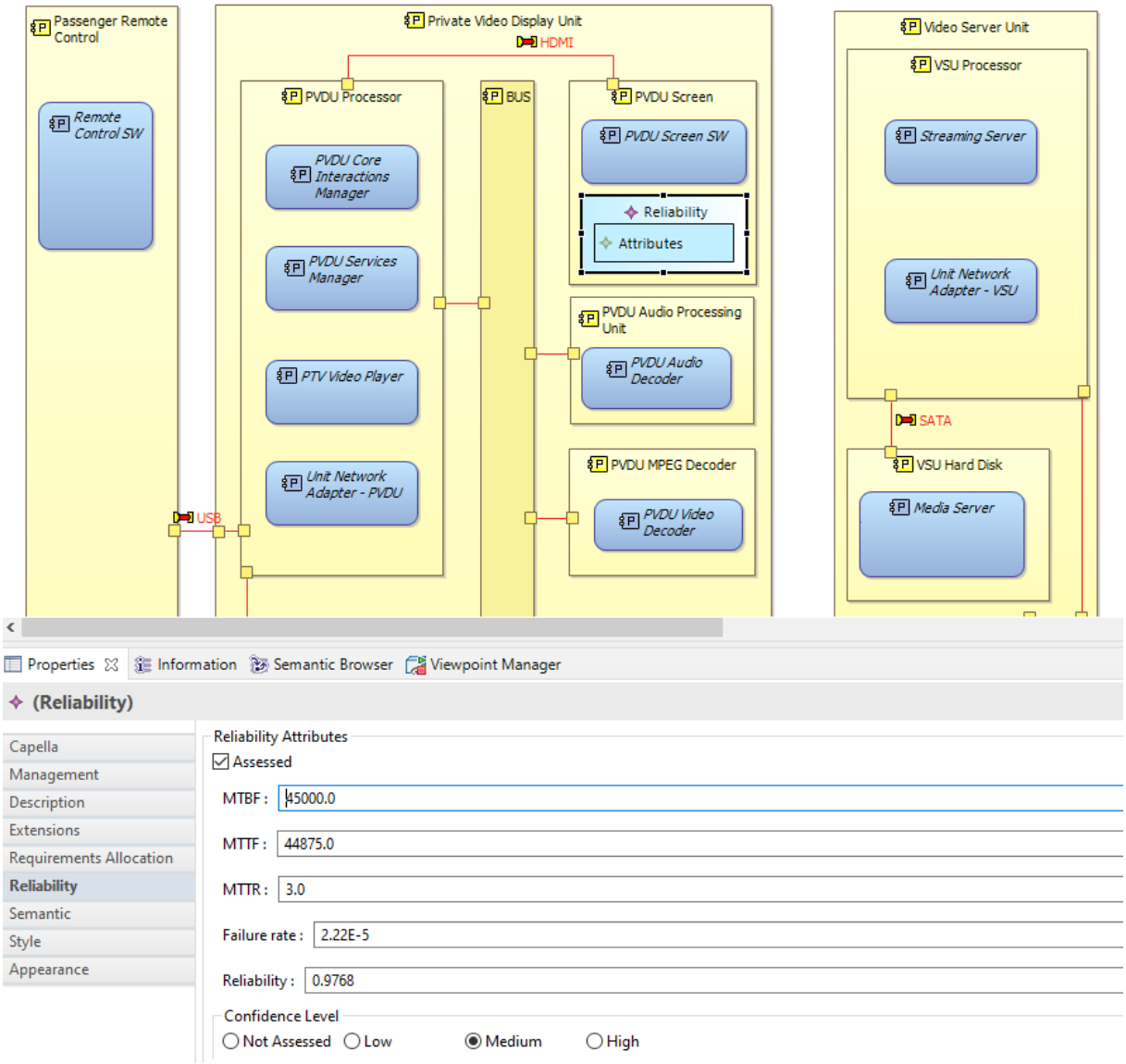

Figure 3. Arcadia/Capella diagram and reliability viewpoint.

It is important to note that our approach does not involve conducting analyses to obtain system reliability data, such as fault tree analysis. It is assumed that the required information, including the specifications of existing components that will be part of a system, is available or has been obtained. Arcadia/Capella diagrams have been integrated with various reliability and safety analysis methods and tools, e.g., ATICA [50]. These methods and tools, and others, can be utilised prior to conducting the System Modelling activity, and their outcomes can be used as inputs for the activity.

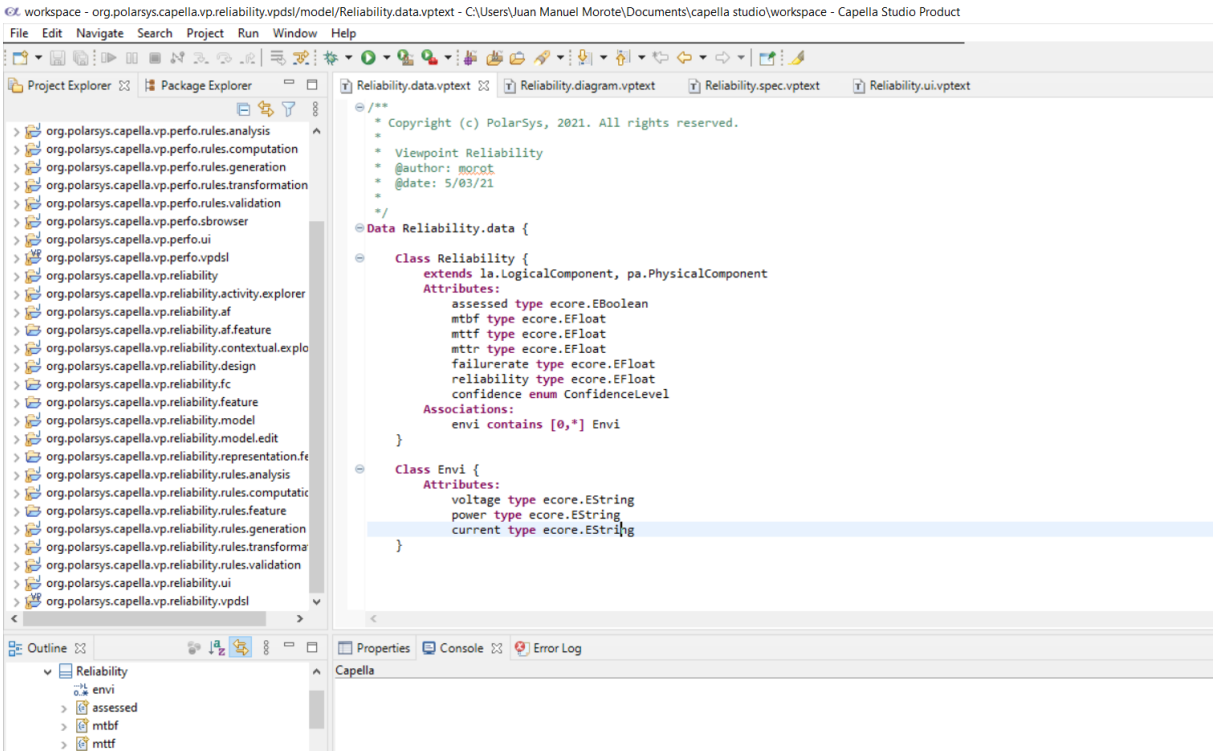

Figure 4. Capella Studio: definition of attributes of a viewpoint.

3.2. Ontology Development

In modern organisations, knowledge has become one of the most valuable assets. Effective management of this asset is crucial for success. This involves gathering knowledge from various sources and storing it in repositories. This knowledge can then be reused as a key asset in systems and software projects. Knowledge Manager supports these needs via storage of knowledge by means of ontologies in well-structured repositories, ensuring easy access and maintenance. The knowledge stored is organised within a System Knowledge Repository and is used by other SES tools.

Ontologies are necessary to apply KCSE with SES, including for AI-based capabilities. Default ontologies are provided by SES for various languages, containing predefined terms, semantic categories, and patterns, among other elements. These ontologies must be customised and expanded to meet the specific needs of system domains and reliability analysis.

Our approach does not define specific needs for the Formalisation and Inference Rules parts of the ontologies used by SES, but only reutilizes their existing elements for purposes such as quality analysis. The rest of the parts must include specific elements:

1. In the Terminology part, the user defines terms related to the domain, such as those pertaining to reliability, as well as those from relevant engineering and assurance standards. System-specific terms can be automatically imported from Arcadia/Capella diagrams.

2. In the Conceptual Model part, the user specifies the semantic categories for terms, such as those related to standards, and specifies the relationship types between system artefacts. All the ontologies created for standards have two clusters, one for the glossary of terms and another for the acronyms (Figure 5).
3. In the Patterns part, the user can add, edit, or delete system specification patterns. The approach focuses on patterns for reliability and safety requirements. A total of 26 sentence patterns have been specified from the literature [69–71]. An example of the patterns is the following: When detecting <failure modes> <within-before-after-exactly at-no later than-every>, <Logical component> should <alarm> <for relevant persona>.

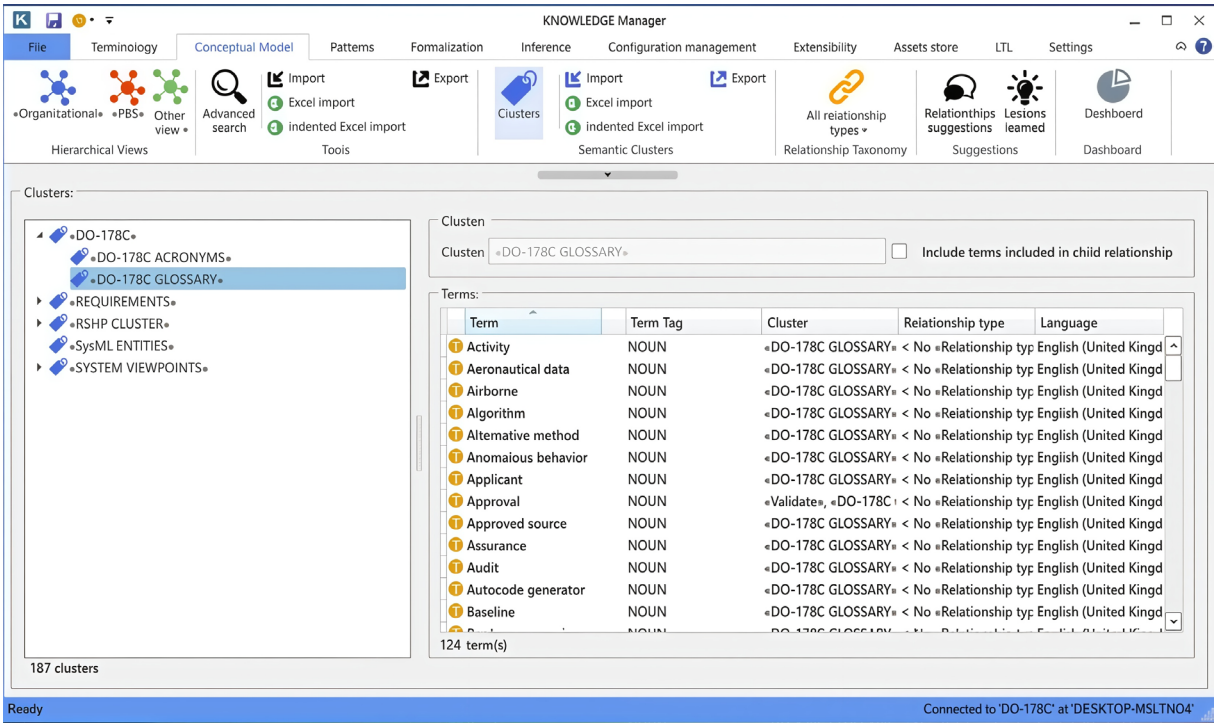

Figure 5. Ontology for the DO-178C standard in Knowledge Manager.

These elements are used in other activities of the approach, e.g., for NLP and reasoning. NLP exploits terms and their syntactic information from the Terminology part, their relationships and semantic information from the Conceptual Model part, and references to terms, to syntactic information, and to semantic information from the Patterns part. Reasoning (Inference Rules part) is mainly based on elements of the other parts of an ontology via procedures that consider these elements.

The ontologies are tailored to specific scopes, such as reliability considerations, system domains, and engineering and assurance standards. SES also provides ontology management functionality [21], including copy and merge, enabling users to jointly utilise content from multiple base ontologies while also managing ontology versions. This functionality also considers possible semantic conflicts when merging ontologies from different domains or overlapping standards, warning users and guiding them in their resolution.

Regarding the IFE system:

1. The DO-178C standard incorporates relevant terms such as ‘configuration item’, ‘high-level requirement’, ‘memory device’, and ‘safety monitoring’, ‘video monitor’ is a term from the Arcadia/Capella diagrams of the system, and failure is a reliability-related term (in addition to a DO-178C one).

2. The 'video monitor' term is classified as a 'component' based on its semantics.
3. The pattern 'The <component characteristic> of the <component> shall be between NUMBER and NUMBER <unit>' can be applied to specify certain characteristics of a component, such as stating that 'The touch temperature of the video monitor shall be between 0 and 49 Celsius degrees'.

3.3. Structured Textual Requirements Specification

Using the RAT tool as a plug-in (Figure 6), an engineer can specify textual requirements on Arcadia/Capella diagrams based on SES ontologies and using NLP. These requirements are linked to diagram elements such as functions of a System Functional Breakdown diagram or components of a Logical Architecture diagram. NLP features analyse the content of textual requirements to determine, e.g., whether they match an established, recommended specification pattern.

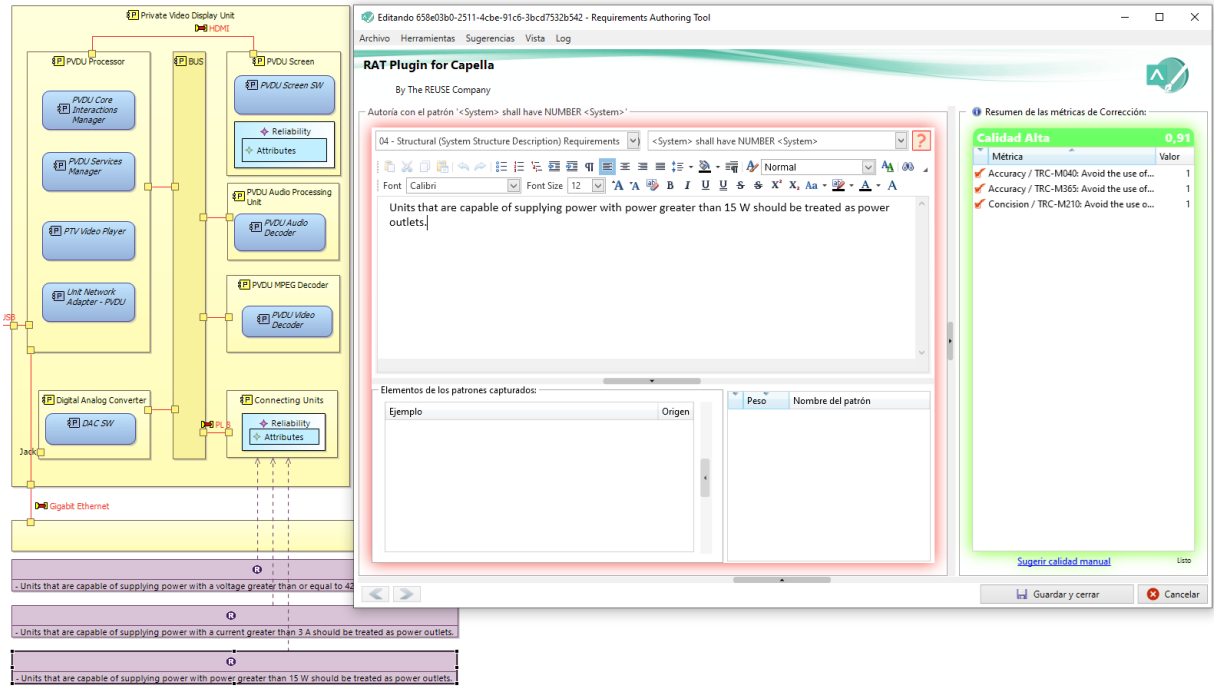

Figure 6. Requirement specification on Capella with RAT.

Writing requirements can be a challenging task that requires familiarity with, e.g., the appropriate vocabulary, specification structures, and measurement units determined for a project, and with quality standards or regulations to conform to. As a solution, RAT is a valuable tool that can assist in creating requirements specifications. By offering a collection of best practices, RAT guides engineers in selecting the most suitable content and ensuring proper grammar for their requirements, according to the Terminology, Conceptual Model, and Patterns of an ontology. Additionally, RAT generates real-time quality reports (based on the Inference Rules part) that identify possible issues in the requirements, reducing the need for time-consuming V&V, peer-review, and double-checking.

The patterns that RAT exploits can be easily tailored and managed through integration with Knowledge Manager, including those patterns defined for reliability and safety requirements during Ontology development. If new patterns are generated and transferred to the production environment, RAT can detect them, allowing authors to use them immediately. Furthermore, RAT can suggest new patterns or modifications to existing ones, alerting knowledge architects who can either approve or reject these suggestions. These features exploit NLP.

By utilising RAT for requirements specification, a structured specification can be developed and aligned with the content of ontologies. This helps to ensure that, e.g., the reliability requirements associated with diagram elements are of high quality. RAT can use off-the-shelf quality metrics (Inference Rules) provided by RQA to perform on-the-fly analysis, e.g., about possible ambiguities, imprecisions, or inconsistencies in a requirement, as a rule-based intelligent system. The metrics can be enabled or disabled as needed, depending on the Capella elements where RAT is being used. Additionally, the use of RAT may lead to the refinement of ontologies, as requirement authors can identify elements that are missing in an ontology, potentially causing quality issues such as the absence of domain concepts. More details about the definition and usage of RQA requirements quality metrics can be found in other publications, e.g., [72].

Reliability-related information can be added through requirements using RAT to the following Arcadia/Capella diagrams and elements:

- Operational Capabilities diagram: Operational Capabilities.
- Operational Activity Interaction diagram: Operational Activities, Interactions.
- Operational Activity Breakdown diagram: Operational Activities.
- Operational Architecture diagram: Operational Activities, Operational Entities, Interactions
- Class diagram: Classes, Associations.
- System Functional Breakdown diagram: System Functions.
- System Architecture diagram: System Functions, System Components, Component Exchanges.
- Logical Functional Breakdown diagram: Logical Functions.
- Logical Component Breakdown diagram: Logical Components.

For the IFE system, RAT can be used to specify requirements for the power, voltage, and current of Connecting Units (physical component), and to associate the requirements with diagram elements. These requirements can be found in rules related to airworthiness of products, parts, and appliances [64]. In addition, according to DO-178C, the requirements should be analysed for ambiguities, inconsistencies, and undefined conditions. RAT can be used for such an analysis.

3.4. Traceability Management

For Traceability management, it may be necessary to handle connections between (a) Arcadia/Capella diagrams or their individual components and (b) other system artefacts of different types (such as requirements or design artefacts) and in various formats (such as models, documents and spreadsheets produced with other tools), as well as between elements of Arcadia/Capella diagrams. This includes traceability management for reliability-related information. The types of relationships can be defined within an ontology's Conceptual Model. Artefact semantics is also considered for traceability purposes.

Traceability Studio enables system engineers to manage various types of system artefacts, ranging from high-level goals and requirements to system or subsystem requirements, risks, and verification actions. In addition, through its connectivity capabilities and based on the Formalisation part of its ontologies, SES can import data from external tools and artefacts, including SysML/UML models, simulation models, MS Excel worksheets, MS Word documents, and requirements managed in various types of repositories (DOORS, Teamcenter, Polarion…) [15]. NLP is used when importing text, e.g., to transform it into SES data format according to patterns. As a result, textual artefacts will be represented in a semantically rich format that does not consider non-meaningful details of the artefacts, such as articles.

Given the large number of different sources involved in a complex systems engineering project, maintaining traceability among all the elements is crucial for project success and often required by standards, guidelines, best practices, or recommendations. Therefore,

SES provides mechanisms to specify all the system artefacts of a project, including models, physical documents, and other types of containers of engineering work products, and to define the different semantics of the traces to be created between the items contained in those elements. This can be accomplished without opening the source tool used to create and manage the items involved. Traceability Studio offers the possibility to create a Module Map, which is a graphical representation of the structure of a traceability project. Traceability Studio also has the capability to suggest new traces by analysing system artefacts (based on the Inference Rules part of an ontology) and allows users to generate an Impact Analysis. The suggestions can consider the terminological, semantic, and structural similarity of the artefacts according to the content of an ontology, among other characteristics. Thresholds can be specified to determine how similar two artefacts must be to suggest that a relationship could exist between them. Trace suggestion is part of the SES inference mechanisms.

As an example, 'realizes' is a relationship type for the IFE system (Figure 7) that can be employed to establish links between Arcadia/Capella diagram elements and reference requirements [64] in an Excel file. These reference requirements may include reliability ones, e.g., "If a component of the IFE system is designed to transmit the necessary safety information (e.g., the passenger briefing), any replacement system should also fulfil the safety objectives mandated for that function." Specific relationship types to consider for compliance with DO-178C include those between system requirements allocated to software and high-level requirements, high-level requirements and low-level requirements, low-level requirements and source code, and software requirements and test cases.

3.5. Model Quality Analysis

Low-quality system artefacts during the concept and design stages of a project can result in rework, additional expenses, delays, and potentially significant consequences if not detected, in addition to system reliability issues. To avoid these problems, RQA is a rule-based intelligent system that automates the process of inspecting and analysing different types of system artefacts, reducing the effort of quality reviews.

To analyse the quality of Arcadia/Capella diagrams in SES, diagram data, including reliability data, can be imported (Formalisation part of an ontology). First, a connection must be established between SES and an ontology that serves as the basis for the analysis. Second, it is necessary to connect Capella and SES. RQA can then utilise data from Arcadia/Capella diagrams (Figure 8) to automate the process of inspecting and evaluating diagram quality. After importing the data, a quantitative analysis of the quality can be performed using various existing metrics and measurement procedures in RQA (Inference Rules part of a SES ontology, based on elements of other parts of the ontology). The content of the Terminology, Conceptual Model, and Patterns of an ontology is also used, as well as NLP, when analysing textual artefacts. NLP functionality can distinguish the different elements of textual artefacts, classify the elements according to their semantics, disregard those elements that do not contribute to establishing semantics (e.g., articles), match the artefacts with patterns, and reason from all these aspects to evaluate artefact quality, among other features.

The evaluations with RQA primarily involve text analysis based on ontology content, such as identifying the use of certain terms and patterns, according to industrial recommendations and best practices. Appendix B lists all the characteristics (30) that are already covered by default in RQA and can be checked for quality analysis of Arcadia/Capella diagrams, such as Ambiguous universal keywords (e.g., "all", "any", or "both"), escape clauses (e.g., "when possible", "if necessary", and "and so on", whose use is an indicator of imprecision, ambiguity, and non-verifiability) and passive voice (as an indicator of

imprecision and ambiguity). Early identification of the associated issues is essential for system reliability. Missing them might result in incorrect or incomplete implementation of required functions, e.g., as a consequence of an ambiguous system specification.

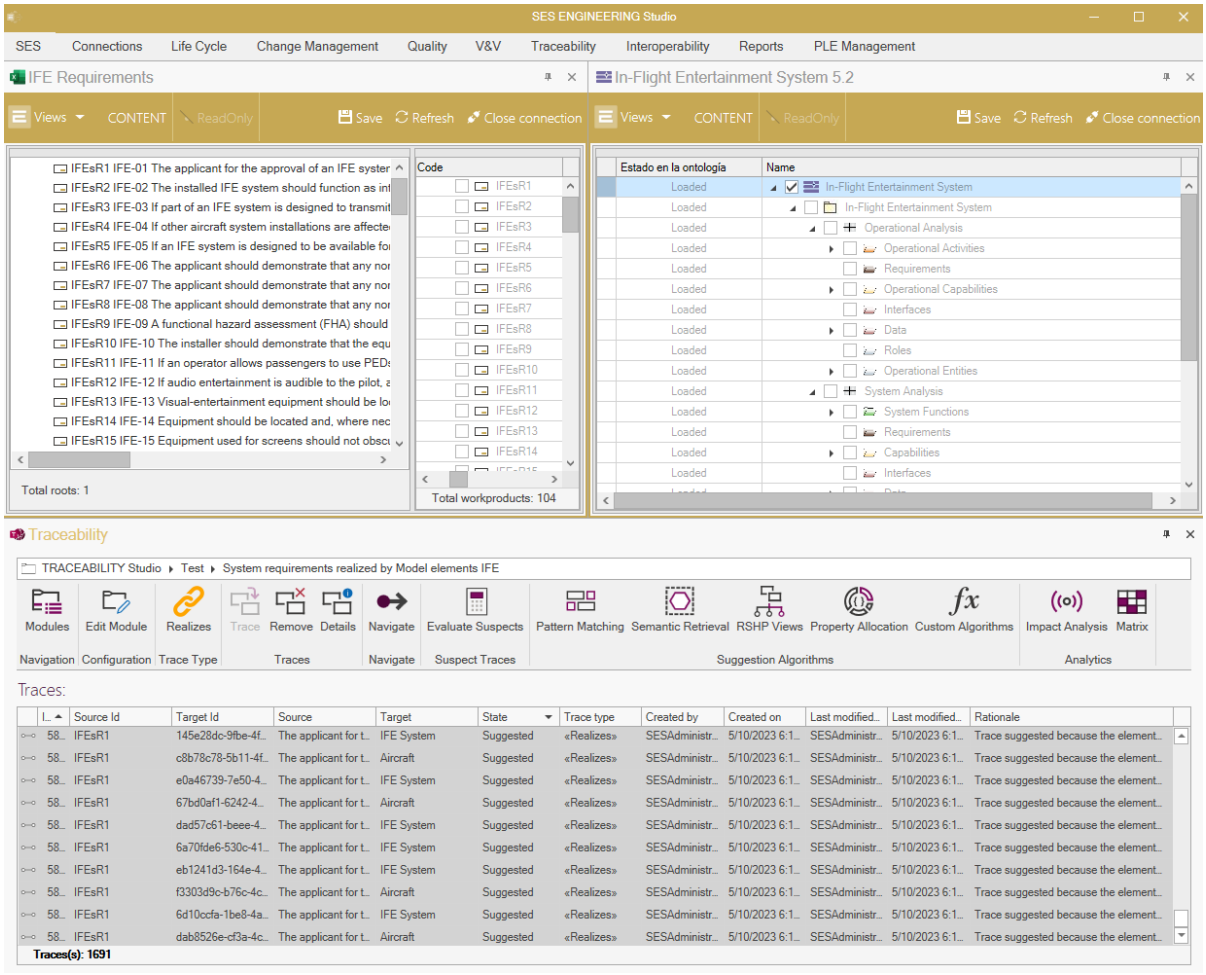

Figure 7. Traceability Studio screenshot.

The information included in the viewpoints can also be analysed to check its consistency, e.g., between the reliability information of a component and the reliability information of its sub-components. To this end, a property consistency metric can be created in RQA to compare the values of (reliability) properties extracted from the elements of Arcadia/Capella diagrams. For example, it must be ensured that the expected failure rate or mean time between failure of a sub-component ensures that the expected failure rate or mean time between failure of its parent component can be achieved.

Users can enable or disable metrics for element types whose quality needs to be evaluated. For instance, when evaluating the quality of an operational capability, it might be necessary to enable metrics that detect the use of passive voice or vague verbs. But if the user wants to measure the quality of a logical or physical component, these metrics will most likely not be useful because of how they are usually named (without verbs). In this case, metrics that, e.g., evaluate readability or check the text length might be enabled.

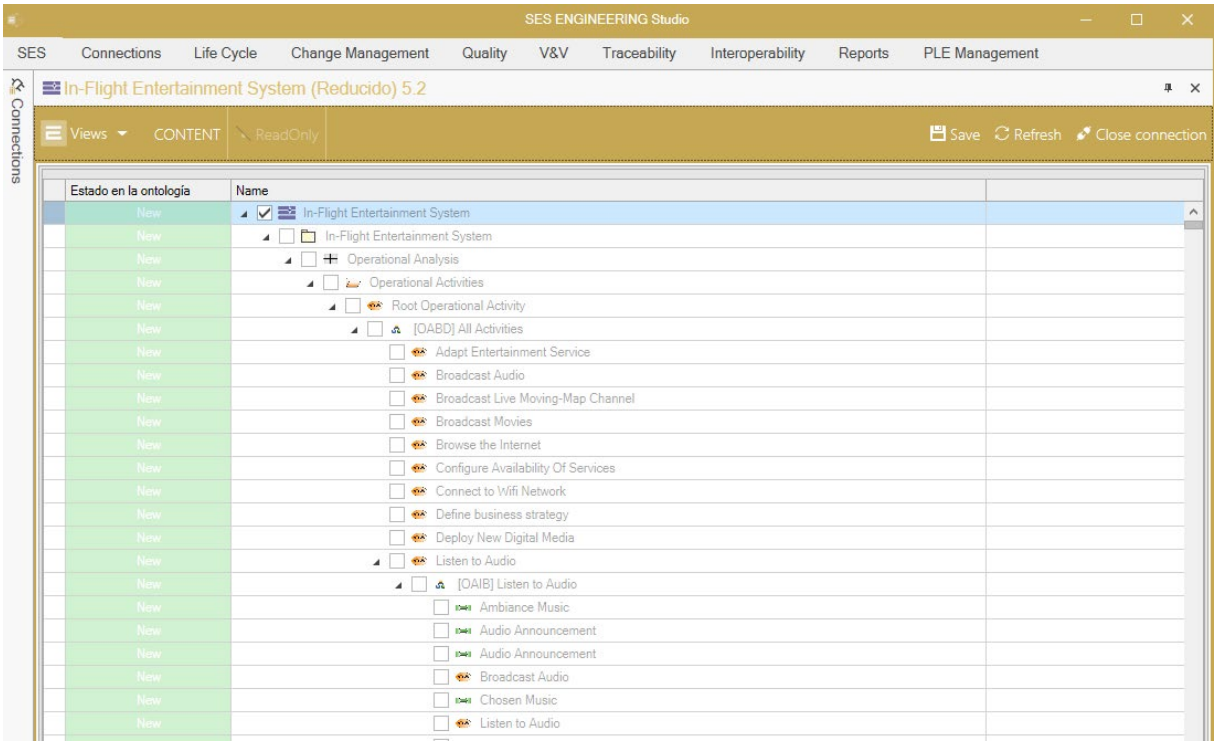

Figure 8. Data from Arcadia/Capella diagrams imported into SES.

For further quality analysis, V&V Studio supports manual assessment by allowing engineers to create and perform other specific V&V actions via various means, e.g., checklists with items related to Arcadia/Capella projects and diagrams. This feature can also be used to create checklists for assessing compliance against engineering and assurance standards, which can serve as assurance evidence [53].

As an example, it is possible to evaluate for the IFE system whether its operational capabilities follow the pattern verb-object, such as 'Provide Video Gaming Services'. For a DO-178C checklist, items to consider may include checking whether interface, performance, and reliability requirements have been specified, among other criteria. Figure 9 shows an overview of quality analysis results for the individual elements of the Arcadia/Capella diagrams of the IFE system. RQA qualitatively rates the quality as low, medium, or high, representing this with one, two, or three stars, respectively. Quality (thus star) rating is based on quantitative quality scores that RQA provides. We use RQA default criteria on what low, medium, and high quality are, e.g., based on the weight assigned to different quality metrics, but users can tailor them. These analyses can contribute to confirming characteristics that DO-178C requires for requirements and architecture artefacts, such as accuracy, consistency, verifiability, and conformance to standards.

Finally, it is important to mention that model quality analysis with RQA complements Capella Validation Rules. Capella provides a diagram validation feature that executes a set of rules to analyse certain characteristics of Arcadia/Capella diagrams, e.g., that the elements have a name, that the names of the elements are not repeated in the same scope, that the elements are associated with others, and the consistency between some elements of the different Arcadia steps. The set of rules to execute can be customised.

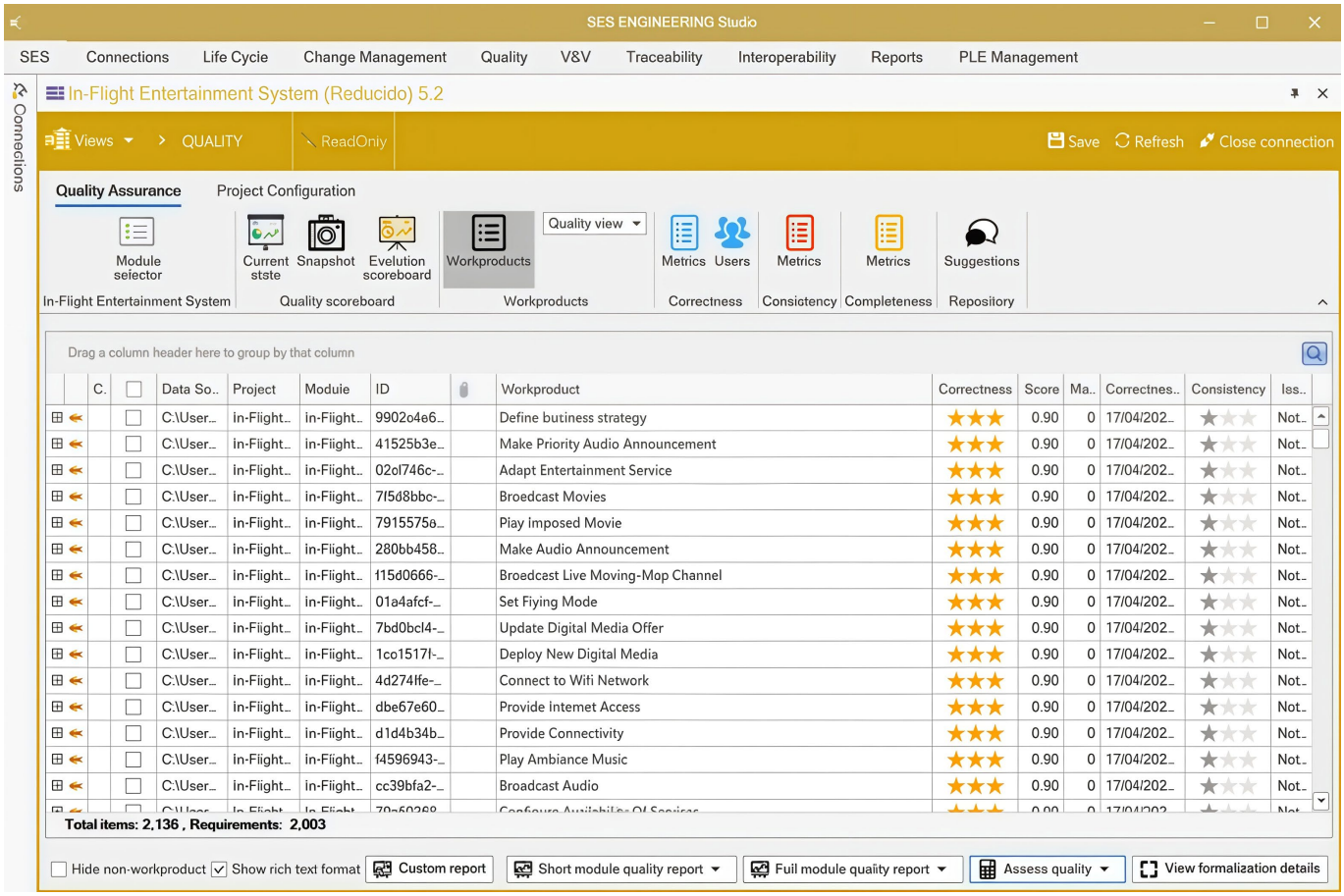

Figure 9. Arcadia/Capella diagram quality analysis with RQA: overall results.

4. Approach Validation

This section presents how we have validated the approach that combines MBSE and KCSE to design reliable systems in practice. We have applied the approach to different existing systems, considering system modelling with Arcadia/Capella and system information management with SES (including AI-based features), and taking into account system, software, and hardware aspects. Such validation can be regarded as case study research [73]. This type of research aims to investigate contemporary phenomena within their real-life context, especially when the boundary between the phenomena and the context cannot be clearly specified. Case study research is typically exploratory and flexible and uses qualitative data as the primary source.

The goal of the validation was to evaluate the effectiveness of the approach that combines MBSE and KCSE to design reliable systems. Two research questions were formulated:

- RQ1: How can the approach be applied to design specific systems?
- RQ2: Can the approach be a feasible means to design reliable systems?

The next sections present the design, the results, and a discussion of the validation.

4.1. Design

The design used to answer the research questions consisted of three elements: (1) selection of relevant and representative systems as MBSE case studies; (2) combination of MBSE and KCSE in the case studies to design reliable systems, and; (3) collection of feedback from practitioners about the combination.

We applied the approach to the following systems:

1. The IFE system (introduced in Section 3).

2. Level Crossing Traffic Control [14]: A level crossing is an intersection where a railway line crosses a road or path, or in rare situations an airport runway, at the same level, as opposed to the railway line crossing over or under using an overpass or tunnel. Level crossings account for many of the catastrophic train accident risks on railways. The reliable and safe design, management, and operation of level crossings can reduce the risks, have a positive effect on user behaviour, and so reduce the number of fatal and serious incidents.
3. Environment Observation Link to Earth (EOLE) [74]: EOLE is a sounding weather balloon system, whose main goal is to provide meteorological data for various scientific users. The EOLE system missions are: (a) to collect weather data (atmospheric pressure, temperature, humidity...) on demand, or according to a predefined schedule; (b) to capture aerial images; (c) to transmit the captured data to a ground station in real-time, and; (d) to propose a subscription mechanism for weather operators.
4. Nutrition Care System [75]: Patients with chronic disease processes are often malnourished on admission to hospital or healthcare facilities. The system deals with the interaction between the patient and various healthcare entities and the definition of patient nutrition care objectives.
5. Arduino Mega2560Rev3 Microcontroller board [76]: It is a microcontroller board with 54 digital input/output pins, 16 analogue inputs, 4 hardware serial ports, a USB connection, a power jack, and a reset button, among other characteristics. The board contains everything needed to support the microcontroller. It can be simply connected to a computer with a USB cable or powered with an AC-to-DC adapter or battery to get started.
6. Hybrid Sports Utility Vehicle (SUV) [76]: A Hybrid SUV car consists of different structural components, such as the engine, body, suspension, gearbox, etc. Its structure, subcomponents, and functions to be performed can be modelled, e.g., the acceleration function, the braking function, the vehicle traction, or how the wheels are assembled.
7. Temperature War [77]: The Temperature War project aims to define, specify, design, build, and run a hardware/software system capable of managing the ambient temperature around itself. The system will operate in the same physical space as other similar systems. The objective of the Temperature War project is the development of a system capable of maintaining its surrounding environment between previously defined thresholds, assuming that the temperature can be affected by the operation of similar systems within its proximity, or by the activity of controlled external agents. It is a system that The REUSE Company (SES developer) uses for testing and product demonstration purposes.
8. Neuromuscular Transmission (NMT) controller [17]: An intelligent infusion controller for vital signs is a medical device that monitors specific vital signs parameters (e.g., blood pressure or NMT) to be regulated. It also infuses, at regular intervals, an updated drug dose value in order to achieve a specific target value for the physiological value under control. This device aims to use a very innovative technology to support the anaesthesiologist in measuring muscle relaxation during an operating room intervention. It corresponds to an industrial use case of the VALU3S project.

The criteria for system selection were availability of diagrams and information, coverage of system, hardware, and software aspects, and coverage of different application domains.

In addition, we considered these standards applicable to the selected systems:

- Automotive: ISO 26262 [4] (Functional safety of road vehicles).

- Avionics: ARP 4754 [78] (Development of civil aircraft and systems), DO-178C [3] (Software of airborne systems and equipment), and DO-254 [79] (Airborne electronic hardware).
- Healthcare: IEC 62304 [80] (Medical device software), ISO 13485 [81] (Quality management systems of medical devices), and ISO 14971 [82] (Application of risk management to medical devices).
- Railway: EN 50126 [83] (Reliability, availability, maintainability, and safety of railway applications), EN 50128 [5] (Software for railway control and protection systems), and EN 50129 [84] (Safety-related railway electronic systems for signalling, communication, signalling, and processing systems).
- The generic IEC 61508 standard [85] (Functional safety of electrical/electronic/programmable electronic safety-related systems).

Finally, we presented the approach and its application to the staff of The REUSE Company, SES users, iRel40 industrial partners, and VALU3S industrial partners.

4.2. Results

Table 1 summarises the application of the approach and its results for the different systems addressed. System modelling, Ontology development, and Model quality analysis were performed for all the systems, whereas Structured textual requirements specification and traceability management were performed for three systems. Different standards have been considered for each system, according to its application domain. The next paragraphs outline the results for each approach activity.

Table 1. Summary of the application of the approach.

	System Modelling	Ontology Dev.	Structured Textual Reqs. Spec.	Traceability Management	Model Quality Analysis	Standards
IFE system	111 diagrams (Viewpoints applicable to 42)	Yes	Yes (applicable to 16 diagrams)	Yes (diagrams and 103 requirements)	Yes	ARP 4754, DO-178C, DO-254,
Level Crossing Traffic Control	102 diagrams (Viewpoints applicable to 46)	Yes	No	No	Yes	EN 50126, EN 50128, EN 50129
EOLE system	29 diagrams (Viewpoints applicable to 12)	Yes	No	No	Yes	ARP 4754, DO-178C, DO-254
Nutrition Care System	13 diagrams (Viewpoints applicable to 5)	Yes	No	No	Yes	ISO 13485, IEC 62304, ISO 14971
Arduino board	21 diagrams (Viewpoints applicable to 20)	Yes	No	No	Yes	IEC 61508
Hybrid SUV	17 diagrams (Viewpoints applicable to 7)	Yes	No	No	Yes	ISO 26262

Table 1. *Cont.*

	System Modelling	Ontology Dev.	Structured Textual Reqs. Spec.	Traceability Management	Model Quality Analysis	Standards
Temperature War	9 diagrams (Viewpoints applicable to 2)	Yes	Yes (applicable to 3 diagrams)	Yes (diagrams and 484 requirements)	Yes	IEC 61508
NMT controller	7 diagrams (Viewpoints applicable to 4)	Yes	Yes (applicable to 3 diagrams)	Yes (diagrams and 42 requirements)	Yes	ISO 13485, IEC 62304, ISO 14971

For System modelling, we considered Arcadia/Capella diagrams and how the viewpoints could be used:

- The diagrams of the IFE system and Level Crossing Traffic Control are publicly available for download [14].
- The diagrams of EOLE and the Nutrition Care systems were reproduced from their main sources ([74] and [75], respectively).
- The diagrams of Arduino and Hybrid SUV were created from available SysML ones [76].
- The diagrams of Temperature War were provided by The REUSE Company.
- The diagrams of the NMT controller were created as a part of the work for the industrial use case of the VALU3S project.

Appendix C lists all the diagrams of the systems. All the Arcadia steps and main diagram types (Section 2.1) were considered, as well as some additional diagram types. For the data in the viewpoints, we either used data from existing component specifications (e.g., [86,87]) or decided upon values that would be representative. Figure 10 shows a Physical Architecture diagram for the Arduino board, including viewpoint data.

In Ontology development, we developed ontologies with the terms, acronyms, and relationship types of the standards taken into account for each system. For example, for the Hybrid SUV, we created an ontology of ISO 26262 with 185 terms, 105 acronyms, a PBS structure (in the Conceptual Model part), and 10 relationship types.

Structured textual requirements specification was addressed for the IFE system (see example in Section 3.3), Temperature War (Figure 11), and NMT controller. We used sets of available system requirements for their specification, analysis, and improvement with RAT, considering specification patterns (available in SES and added) and requirements quality metrics (e.g., about the use of will, must, or shall in the text of the requirements and the use of vague terms).

Similar to the previous activity, traceability management dealt with Arcadia/Capella diagrams and textual requirements (in Excel) of the IFE system, Temperature War, and NMT controller (Figure 12). We specified traces for them, considering the trace types (relationship types in Knowledge Manager) indicated in the standards. In addition, we used Traceability Studio functionality for automated trace discovery and for change impact analysis.

The Arcadia/Capella diagrams of the systems were imported into SES for Model quality analysis. The results from the analysis with existing RQA metrics indicated that the overall quality of the diagrams was high. Nonetheless, a few possible issues were detected, e.g.:

- The length of the description of most of the diagram elements was not adequate (no description).
- In Level Crossing Traffic Control, the comment "A train leaving should not be delayed more than 5 mn" uses "should" (imprecise modal verb), "not" (ambiguous), passive voice (imprecise indication), and "mn" (not in the ontology as a measurement unit).
- In Temperature War, the operational capability "Temperature Regulation" does not follow the pattern verb + object.

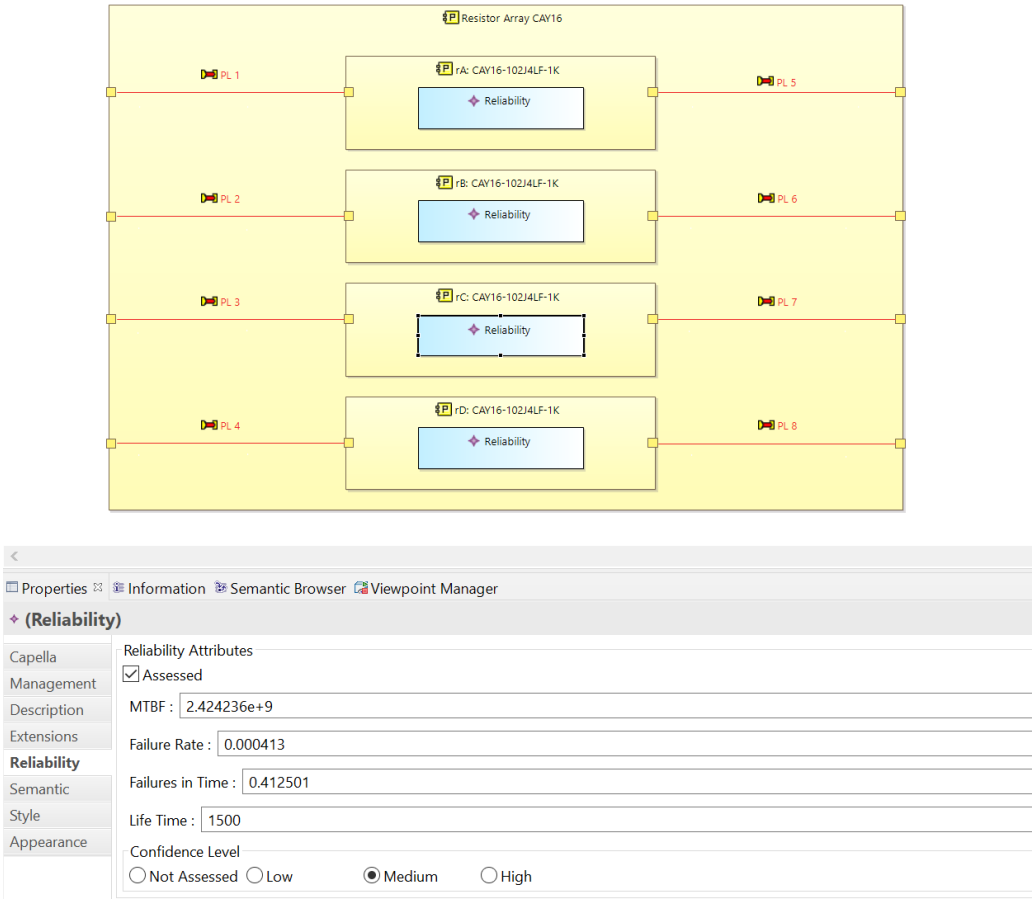

Figure 10. Physical Architecture diagram and viewpoint data for the Arduino board.

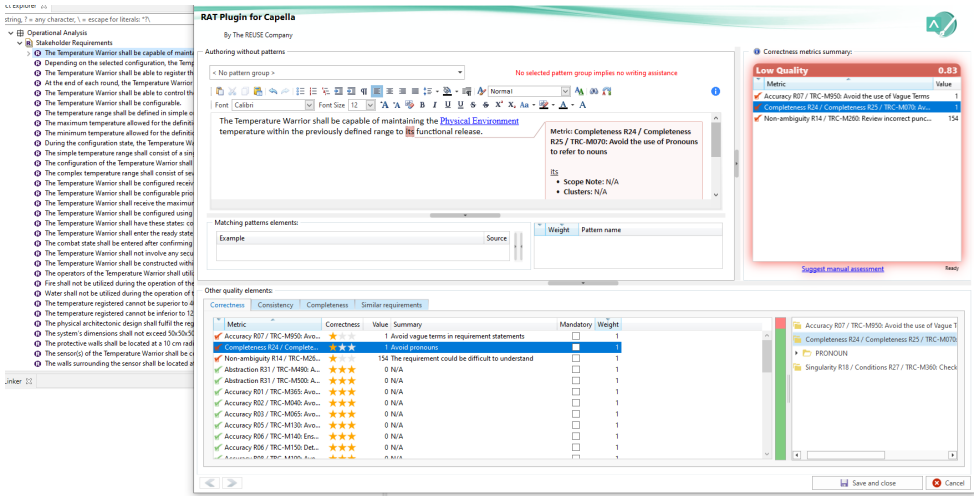

Figure 11. Requirements specification with RAT for Temperature War.

Regarding the checklists, we analysed the coverage of the items identified in the standards related to the requirements and design system lifecycle phases, including requirements and design verification.

Finally, for feedback collection, the approach and its application were presented to tens of practitioners at different events and meetings. Staff of The REUSE Company, SES users, and iRel40 and VALU3S industrial partners provided positive feedback on the feasibility of the approach. Consultants and users of Arcadia/Capella and of SES confirmed the alignment of the approach with how the solutions were used, acknowledging that their combination could be beneficial to design reliable systems. In addition, the practitioners showed particular interest in how Capella had been extended, as well as in the integration and execution of the different activities of the approach in SES. They also asked about how the approach could be adapted to specific project needs and settings, such as the execution of some systems engineering or compliance management tasks with other tools.

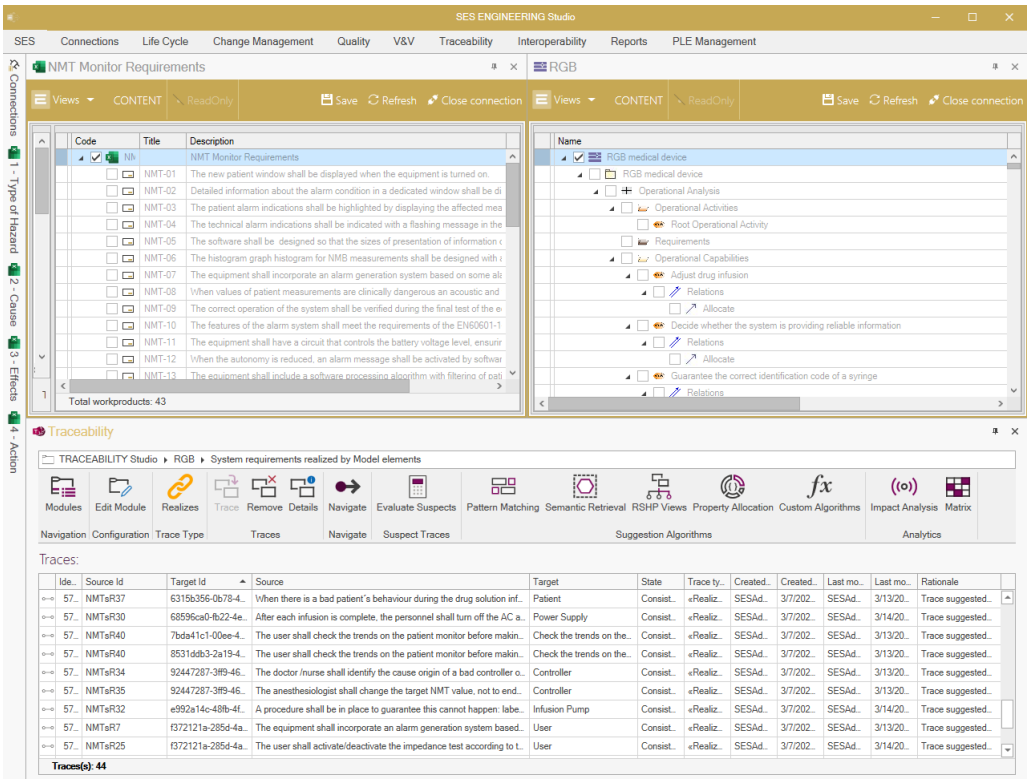

Figure 12. Traceability management with Traceability Studio for the NMT controller.

4.3. Discussion

This section discusses the answers to the research questions formulated, the validity of the approach, and how it has been applied.

4.3.1. Application of the Approach (RQ1)

We have successfully applied the approach on eight systems that cover five different application domains (aerospace, automotive, healthcare, railway, and generic system) and 11 standards, considering tens of diagrams, of knowledge elements, of reliability properties, and of analysis possibilities. System, software, and hardware reliability aspects have been taken into account. Based on the data available about the systems, some activities of the approach have been performed for all the systems, and others only for a few. The systems and their components have been modelled with Capella, and reliability information has been specified via viewpoints. Ontologies have been developed with Knowledge Manager from the knowledge available in standards. Structured textual requirements have been

added to Arcadia/Capella diagrams with RAT. Traceability between Arcadia/Capella diagrams and requirements in spreadsheets has been managed with Traceability Studio. Finally, the quality of Arcadia/Capella diagrams has been analysed, and possible quality issues have been identified with RQA and V&V Studio.

For the application of the approach, it is important to note that it might vary among systems and projects. Some activities of the approach might be executed in a different way or not executed. For example, depending on the needs of the systems and projects, compliance with standards might not be required. This affects Ontology development, Traceability management, and Model quality analysis, which use or can use knowledge from standards. Depending on the settings of the systems or projects, other tools might be used for, e.g., textual requirements specification or traceability management (DOORS, Polarion, PTC Integrity, Reqtify...). Such variations and adaptations are common needs and have been discussed for systems engineering toolchains (e.g., [2]).

4.3.2. Feasibility of the Approach (RQ2)

We argue that the approach can be a feasible means to design reliable systems. As discussed above, we have been able to apply it to different systems, considering their reliability characteristics. In addition, practitioners' feedback has been positive in relation to approach feasibility and to its link with their MBSE or KCSE practices. These practitioners came from different sources, e.g., The REUSE Company and the VALU3S and iRel40 projects.

Nonetheless, some aspects of the feasibility of the approach deserve some further discussion. There are important dependencies with the knowledge about and expertise in MBSE with Arcadia/Capella and KCSE with SES. For example, effective quality analysis with RQA might not be straightforward for novice users because of the background that it requires about ontology development and usage, and about quality analysis configuration. These aspects need to be considered so that the approach is actually feasible. In addition, the actual reliability of the systems would need to be confirmed at operation time. The approach supports the execution of specification and analysis actions at early development stages to ensure that a system will be reliable. Regarding effort and costs in applying the approach, SES usage requires an additional engineering effort when compared to other tools, as well as specific training. Nonetheless, such aspects are expected and even required when engineering, e.g., safety-critical systems, for which ensuring the quality of the systems (including their reliability) and of system specifications is a must. This is necessary for compliance with engineering and assurance standards. Overall, the approach and its application are aligned with what practitioners in general and users of Arcadia/Capella and of SES in particular regard as cost-effective practices.

Last but not least, when comparing the approach and its application with existing solutions, quantitative differences exist. Among them, it can be highlighted that such solutions have not considered:

1. A large set of reliability and environment aspect attributes when modelling systems (24 in our approach).
2. The extent to which various system-, reliability-, and standard-specific terms, semantic aspects, and patterns must be considered in ontology development for reliable-system design (tens of elements when applying our approach).
3. How many types of Arcadia/Capella diagrams and elements can benefit from their integration with RAT for reliability requirement specification (nine diagrams and 15 diagram elements in our approach)?

4. Different artefact types that can be used for advanced traceability management towards system reliability (tens of types in our approach, including all the types of Arcadia/Capella diagrams and elements).
5. The identification of specific metrics for reliability-oriented model quality analysis (30 in our approach).

Therefore, we consider that our approach is a more feasible means than prior work in order to design reliable systems in practice in the context of MBSE and KCSE, and more specifically, combining Arcadia/Capella and SES.

4.3.3. Validity

We discuss validity according to the aspects proposed by Runeson et al. for case study research [73].

We consider that construct validity is largely ensured because of the adequacy of the systems on which the approach that combines MBSE and KCSE to design reliable systems was applied. They correspond to real and representative systems whose reliability must be ensured. The standards considered are also relevant in industry. The threat of mono-method bias could be addressed in the future by conducting experiments that study the use and benefits of the approach, e.g., more accurate requirements specification and diagram quality analysis.

An aspect that affects internal validity is that we, the researchers, were mainly responsible for approach application. We have background on MBSE and KCSE, which impacts the results. They would most likely be different for novice users, who would need extensive training on MBSE, KCSE, Arcadia/Capella, and SES to reach our level of expertise. A key aspect to effectively use SES is to understand how ontologies are developed (e.g., considering different parts) and later used for different KCSE tasks. Another threat is the existence of a relationship with the practitioners who provided feedback on the approach, which can result in a more positive (or more negative) opinion about the approach based on their knowledge about and experience with our work. The opinions might also be different from practitioners who apply the approach as a whole, instead of, e.g., practitioners with great experience, mostly with some parts, such as system modelling with Arcadia/Capella or KCSE with SES.

In general, case study research does not aim to broadly generalise its findings, which impacts external validity. However, our findings are expected to be applicable to system design in situations similar to those in the systems considered, e.g., for systems with similar characteristics and from the same application domains. We also argue that the broad scope of the application of the approach, considering different systems, standards, and application domains, contributes to external validity. It is not common for papers report the validation of some technology with eight systems. Another aspect that contributes to external validity is the fact that Arcadia/Capella and SES have been and are used in different projects, systems, and application domains.

Regarding reliability, our involvement in the approach application affects it. Other people might obtain different results, e.g., when modelling the systems. Nonetheless, we also consider that the degree of detail provided when describing the approach and its application contributes to reproducibility. The lack of a more structured way for feedback collection, e.g., through a predefined questionnaire, affects reliability. Higher priority was assigned to ease feedback provision by practitioners with less formal means, e.g., at meetings of the VALU3S and iRel40 projects with all the partners.

5. Conclusions

It is a must nowadays in many application domains to ensure the reliability of software-intensive systems and to start addressing it at early system lifecycle phases. As support, we have described an approach that combines two industrial practices: MBSE with Arcadia/Capella and KCSE with SES. The approach further exploits AI via knowledge management, NLP, and reasoning capabilities, enabling ontology-driven representation of system and reliability knowledge, automated inference, and semantic traceability analysis.

The approach consists of five activities: (1) System modelling with Arcadia/Capella; (2) ontology development with SES; (3) structured textual requirements specification on Capella with SES; (4) traceability management with SES (including traceability of Arcadia/Capella elements), and; (5) model quality analysis for Arcadia/Capella diagrams with SES. The activities take different pieces of reliability information into account, enabling and supporting the specification and analysis of this information and its association with different elements. Specifically, engineers can (1) use two new viewpoints to specify reliability data in Arcadia/Capella diagrams, (2) include reliability-related concepts and relationships in ontologies, (3) specify textual reliability requirements in a structured way and integrate them into Arcadia/Capella diagrams, (4) create links between reliability-related information and other system artefacts, and (5) conduct model quality analyses that consider reliability aspects. This can be regarded as new reliable system design principles. The approach also addresses compliance with standards, mostly from the development of ontologies that include knowledge from the standards.

We have validated the approach with eight systems from different application domains. We have also presented the approach and its application to tens of practitioners and collected their feedback, which has been positive. These actions have allowed us to show how the proposed combination of MBSE with Arcadia/Capella and KCSE with SES can be used to design specific systems and that the combination can be a feasible means to design reliable systems in practice.

Thanks to the approach, MBSE with Arcadia/Capella and KCSE with SES can benefit from each other, and specifically for reliable-system design. Arcadia/Capella can benefit from new viewpoints, from ontologies, and from AI features in different systems engineering tasks, whereas SES can benefit from the integration with a model-based means whose adoption is growing and whose diagrams and diagram elements can be used in different KCSE tasks. Arcadia/Capella and SES are generic engineering environments applicable to different purposes, such as security-critical system engineering. The approach enacts a feasible combined use of MBSE and KCSE to design reliable systems in practice. This is arguably the main outcome of our work, in addition to the validation results and the corresponding evidence of the effectiveness of the approach. No prior work has effectively combined Arcadia/Capella and SES for reliable-system design, has addressed specification and analysis of reliability needs with MBSE and KCSE to such a large extent, or has validated its results with so many different systems. This includes validation with both well-known, reference industrial systems (e.g., IFE systems) and real ones (e.g., the NMT controller). The approach further builds on mature industrial solutions and provides new means aligned with how the solutions are already used for large, complex critical systems.

In the future, it would be useful to define new, specific quality metrics for Arcadia/Capella diagrams and to implement them in SES. It would also be interesting to study the application of the approach on further systems, especially in collaboration with Capella users and SES users, and considering third-party feedback or independent user studies. The design and execution of experiments that provide further evidence of the benefits of the approach, e.g., for reliability issue detection with RQA vs. via traditional manual reviews, would be valuable as well. Other recommendations for future research include

the application of different AI techniques (e.g., generative AI or agents) in the combination of MBSE and KCSE and the extension of the approach towards solutions that address and analyse reliability in other lifecycle stages (e.g., at runtime) and with further means (e.g., digital twins).

Author Contributions: Conceptualization, J.M.M., J.L.d.l.V., G.G., C.A. and L.A.; Methodology, J.M.M., J.L.d.l.V., G.G., C.A. and L.A.; Software, J.M.M., J.L.d.l.V. and L.A.; Validation, J.M.M., J.L.d.l.V., G.G., C.A. and L.A.; Formal Analysis, J.M.M., J.L.d.l.V., G.G., C.A. and L.A.; Investigation, J.M.M., J.L.d.l.V., G.G., C.A. and L.A.; Resources, J.M.M., J.L.d.l.V., G.G., C.A. and L.A.; Data Curation, J.M.M., J.L.d.l.V. and L.A.; Writing—Original Draft Preparation, J.M.M. and J.L.d.l.V.; Writing—Review and Editing, J.M.M., J.L.d.l.V., G.G., C.A. and L.A.; Visualisation, J.M.M., J.L.d.l.V., G.G., C.A. and L.A.; Supervision, J.L.d.l.V. and L.A.; Project Administration, J.M.M., J.L.d.l.V., G.G., C.A. and L.A.; Funding Acquisition, J.L.d.l.V. All authors have read and agreed to the published version of the manuscript.

Funding: The work leading to this paper has received funding from the following projects: iRel40 (H2020-ECSEL ref. 876659; MCIN/AEI/10.13039/501100011033 ref. PCI2020-112240; NextGenerationEU/PRTR), VALU3S (H2020-ECSEL ref. 876852; MCIN/AEI/10.13039/501100011033 ref. PCI2020-112001; NextGenerationEU/PRTR), REBECCA (HORIZON-KDT ref. 101097224; MICINN/AEI/10.13039/501100011033 ref. PCI2022-135043-2; NextGenerationEU/PRTR), Music360 (Horizon Europe ref. 101094872), AETERNAL (MCIN/AEI/10.13039/501100011033 ref. PID2023-149753OB-C21; ERDF), FDT4S (JCCM ref. SBPLY/24/180225/000020; ERDF), CoMoDiD (Generalitat Valenciana CIPROM/2021/023), and "Una propuesta integral para el desarrollo independiente de dominio de gemelos digitales" (UCLM ref. 2025-GRIN-38441; ERDF). The Ramon y Cajal Program (MCIN/AEI/10.13039/501100011033 ref. RYC-2017-22836; ESF) has also funded the work.

Data Availability Statement: All the research data is not publicly available due to intellectual property reasons.

Acknowledgments: We are grateful to the staff of The REUSE Company, SES users, and iRel40 and VALU3S industrial partners who provided input for and feedback on the approach.

Conflicts of Interest: Author Luis Alonso was employed by the REUSE Company. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Reliability Attributes and Environmental Aspect Attributes for System Modelling

Reliability attributes considered:

- Reliability: probability of success at time t, which is denoted R(t).
- MTBF: Mean Time Between Failure.
- MTTF: Mean Time to Failure.
- MTTR: Mean Time to Repair.
- FIT: Failures in Time.
- Failure Rate: failure/t.
- MTBMA: Mean Time Between Maintenance Actions.
- MTBUR: Mean Time Between Unscheduled Removal.
- Availability: probability that the system is applicable for use at a given time.
- Maintainability: probability that a system or system element can be repaired in a defined environment within a specified period of time.
- ROCOF: Rate of occurrence of failure. The number of unexpected events over a specific time of operation.
- POFOD: Probability of Failure on Demand. Possibility that the system will fail when a service request is made.

- PPC: Planned Maintenance Percentage. Percentage of time spent on planned maintenance in contrast to unplanned maintenance tasks. PPC = (Planned maintenance time/Total maintenance time) × 100.
- Probability of survival, for a specific period of time t.
- Probability of success, independent of time.
- Life Time.

Environmental aspect attributes considered:

- Temperature
- Humidity
- Voltage
- Vibration
- Weather (wind, rain, snow)
- Pressure
- Ambient Light
- Operator Skills

Appendix B. Quality Analysis Characteristics

Characteristics that can be checked for quality analysis of Arcadia/Capella diagrams:

- Ambiguous universal keywords, e.g., "all", "any", or "both".
- Combinators out of the condition block, because the use of, e.g., "and", "or", "unless", and "but also" can be ambiguous and show lack to atomicity.
- Convention for logical expression forms, because the use of "and/or" is ambiguous.
- Element description length, for clarity and understandability.
- Element name length, for clarity and understandability.
- Escape clauses, such as "when possible", "if necessary", and "and so on", whose use is an indicator of imprecision, ambiguity, and non-verifiability.
- Flow sentences, because terms such as "although", "as well as", "except", and "unless" should be avoided to ensure the consistency of a specification.
- Imprecise modal verbs, because the use of "should", "might", and "may" might result in imprecise and ambiguous sentences.
- Imprecise quantifiers, such as "a lot of", "hundreds of", "portions of", and "plenty of".
- Inadequate unit for a characteristic, because otherwise a specification would be inaccurate.
- Incorrect punctuation, for non-ambiguity.
- Incorrect spelling, for non-ambiguity.
- Indefinite articles in front of an agent, because "a" and "an" should not be used for accuracy.
- Mixing up different measurement systems, which indicate imprecision and inconsistency.
- Multiple verbs, which show lack of atomicity.
- 'Not' and other negative expressions, because they can result in confusion.
- Numbers followed by units or noun qualifications, for accuracy.
- Open-ended clauses, e.g., "TBD", "etc.", and "including but not limited to", which are imprecise, ambiguous, and non-verifiable.
- Parentheses, as an indicator of imprecision and ambiguity.
- Passive voice, as an indicator of imprecision and ambiguity.
- Pattern matching metric for Noun structure, to ensure the use of suitable name structures for certain Arcadia/Capella diagram elements.
- Pattern matching metric for Verb + Object, to ensure the use of suitable name structures for certain Arcadia/Capella diagram elements.
- Phrases that indicate the purpose, such as "in order to", "so that", and "thus allowing", which can be confusing.

- Pronouns to refer to nouns, which can indicate ambiguity and be inconsistent.
- Readability, as an indicator of ambiguity.
- Superfluous infinitives, e.g., "be capable of" and "be designed to", as an indicator of possible issues in clarity and understandability.
- Synonyms, because the use of the main term instead of synonyms contributes to precision and consistency.
- Temporal indefinite keywords out of the condition part, because terms such as "finally", "almost always", "in the end", or "ultimately" can cause confusion or unintended meaning.
- Unachievable absolutes expressions, i.e., unrealistic, absolute, and non-verifiable expressions such as "100% availability", "always", and "never".
- Vague terms, including adjectives (e.g., "small", "typical", "appropriate", and "common"), adverbs (e.g., "usually", "approximately", and "sufficiently"), and verbs (e.g., "assist", "facilitate", and "manage") whose use can lead to imprecision, ambiguity, and non-verifiability.

Appendix C. Diagrams for Approach Validation

IFE System (111 diagrams)

- Operational Analysis
  - 1 Operational Entity Breakdown diagram
  - 1 Operational Capabilities diagram
  - 1 Operational Activity Breakdown diagram
  - 3 Operational Activity Interaction diagrams
  - 2 Operational Process Description diagrams
  - 2 Operational Architecture diagrams
  - 1 Operational Entity Scenario diagram
- System Analysis
  - 5 Model State Machine diagrams
  - 1 Contextual System Actors diagram
  - 2 System Functional Breakdown diagrams
  - 9 System Functional Dataflow diagrams
  - 5 System Functional Chain Descriptions
  - 6 Functional Scenario diagrams
  - 2 System Architecture diagrams
  - 6 Exchange Scenario diagrams
  - 1 Missions Capabilities diagram
  - 1 Contextual Detailed Interfaces diagram
  - 3 Class diagrams
  - 2 Contextual Capability diagrams
  - 2 Contextual Mission diagrams
- Logical Architecture
  - 1 Logical Functional Breakdown diagram
  - 10 Logical Functional Dataflow diagrams
  - 5 Logical Functional Chain Description diagrams
  - 1 Logical Component Breakdown diagram
  - 6 Logical Architecture diagrams
- Physical Architecture
  - 1 Physical Functional Breakdown diagram

- 11 Physical Functional Dataflow diagrams
- 6 Physical Functional Chain Description diagrams
- 1 Physical Path Description diagram
- 2 Physical Component Breakdown diagrams
- 10 Physical Architecture diagrams

- End-Product Breakdown Structure
  - 1 EPBS Architecture diagram

Level Crossing Traffic Control (102 diagrams)

- Operational Analysis
  - 1 Operational Entity Breakdown diagram
  - 1 Operational Capabilities diagram
  - 3 Operational Process Description diagrams
  - 4 Operational Architecture diagrams
  - 5 Operational Entity Scenario diagrams
  - 1 Capability Realisation diagram
- System Analysis
  - 5 Model State Machine diagrams
  - 2 System Functional Breakdown diagrams
  - 9 System Functional Dataflow diagrams
  - 9 System Functional Chain Description diagrams
  - 1 Functional Scenario diagram
  - 9 System Architecture diagrams
  - 3 Exchange Scenario diagrams
  - 1 Missions Capabilities diagram
  - 1 Contextual External Interface diagram
  - 9 Class diagrams
- Logical Architecture
  - 2 Logical Functional Breakdown diagrams
  - 5 Logical Functional Dataflow diagrams
  - 1 Logical Functional Chain Description diagram
  - 8 Logical Architecture diagrams
- Physical Architecture
  - 1 Physical Functional Breakdown diagram
  - 2 Physical Functional Dataflow diagrams
  - 18 Physical Architecture diagrams
- End-Product Breakdown Structure
  - 1 Configuration Items Breakdown diagram

EOLE system (29 diagrams)

- Operational Analysis
  - 1 Operational Entity Breakdown diagram
  - 1 Operational Capabilities diagram
  - 3 Operational Activity Interaction diagrams
  - 1 Operational Architecture diagram
  - 1 Operational Entity Scenario diagram
- System Analysis
  - 1 System Functional Breakdown diagram

- 3 System Functional Dataflow diagrams
- 2 System Functional Chain Description diagrams
- 1 Functional Scenario diagram
- 2 System Architecture diagrams
- 1 Exchange Scenario diagram
- 1 Contextual Detailed Interfaces diagram
- 1 Contextual External Interface diagram
- 1 Class diagram
- 1 Model State Machine diagram

- Logical Architecture
  - 1 Logical Architecture diagram
- Physical Architecture
  - 1 Physical Functional Breakdown diagram
  - 1 Physical Functional Dataflow diagram
  - 1 Physical Functional Chain Description diagram
  - 1 Physical Path Description diagram
  - 1 Physical Architecture diagram
- End-Product Breakdown Structure
  - 1 Configuration Items Breakdown diagram
  - 1 EPBS Architecture diagram

Nutrition Care System (13 diagrams)

- Operational Analysis
  - 1 Operational Entity Breakdown diagram
  - 2 Operational Capabilities diagrams
  - 2 Operational Activity Interaction diagrams
  - 2 Operational Process Description diagrams
  - 1 Operational Architecture diagram
- System Analysis
  - 1 Model State Machine diagram
  - 1 System Functional Dataflow diagram
  - 1 System Architecture diagram
  - 1 Missions Capabilities diagram
- Logical Architecture
  - 1 Logical Architecture diagram

Arduino Mega2560Rev3 Microcontroller board (21 diagrams)

- Logical Architecture
  - 1 Logical Component Breakdown diagram
  - 3 Logical Architecture diagrams
- Physical Architecture
  - 17 Physical Architecture diagrams

Hybrid Sports Utility Vehicle (17 diagrams)

- Operational Analysis
  - 2 Operational Capabilities diagrams
  - 2 Operational Activity Interaction diagrams
- System Analysis

- 7 Model State Machine diagrams
- 1 Logical Component Breakdown diagram
- 5 Logical Architecture diagrams

Temperature War (9 diagrams)

- Operational Analysis
  - 1 Operational Entity Breakdown diagram
  - 1 Operational Capabilities diagram
- System Analysis
  - 1 Contextual System Actors diagram
  - 1 System Functional Breakdown diagram
  - 1 Missions Capabilities diagram
  - 1 Model State Machine diagram
- Logical Architecture
  - 1 Logical Component Breakdown diagram
  - 1 Logical Architecture diagram
- Physical Architecture
  - 1 Physical Architecture diagram

NMT controller (7 diagrams)

- Operational Analysis
  - 1 Operational Capabilities diagram
- System Analysis
  - 1 System Architecture diagram
  - 1 Model State Machine diagram
- Logical Architecture
  - 1 Logical Component Breakdown diagram
  - 3 Logical Architecture diagrams

References

1. de la Vara, J.L.; Bauer, T.; Fischer, B.; Karaca, M.; Madeira, H.; Matschnig, M.; Mazzini, S.; Nandi, G.S.; Patrone, F.; Pereira, D.; et al. A Proposal for the Classification of Methods for Verification and Validation of Safety, Cybersecurity, and Privacy of Automated Systems. In Proceedings of the 14th International Conference on the Quality of Information and Communications Technology (QUATIC 2021), Algarve, Portugal, 8–11 September 2021.
2. de la Vara, J.L.; Ruiz, A.; Blondelle, G. Assurance and Certification of Cyber-Physical Systems: The AMASS Open Source Ecosystem. *J. Syst. Softw.* **2021**, *171*, 110812. [CrossRef]
3. *DO-178C*; Software Considerations in Airborne Systems and Equipment Certification. RTCA: Washington, DC, USA, 2012.
4. *ISO 26262*; Road Vehicles—Functional Safety. 2nd ed. ISO: Geneva, Switzerland, 2018.
5. *EN 50128*; Railway Applications—Communications, Signalling and Processing Systems—Software for Railway Control and Protection Systems. 2nd ed. CENELEC: Brussels, Belgium, 2011.
6. *ISO/IEC/IEEE 24765*; Systems and Software Engineering—Vocabulary. 2nd ed. ISO: Geneva, Switzerland, 2017.
7. Holt, J.; Perry, S.; Brownsword, M. *Foundations for Model-Based Systems Engineering: From Patterns to Models*; IET: Stevenage, UK, 2016.
8. Parra, E.; Alonso, L.; Mendieta, R.; de la Vara, J.L. Advances in Artefact Quality Analysis for Safety-Critical Systems. In Proceedings of the 30th International Symposium on Software Reliability Engineering (ISSRE 2019), Berlin, Germany, 27–30 October 2019.
9. Russel, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*, 4th ed.; Pearson: Harlow, UK, 2021.
10. Al-Sharif, Z.A.; Jeffery, C.L. AbstractTrace: The Use of Execution Traces to Cluster, Classify, Prioritize, and Optimize a Bloated Test Suite. *Appl. Sci.* **2024**, *14*, 11168. [CrossRef]

11. Maru, G.G.; Lee, S.; Ji, S.; Ko, S.K.; Im, H. Neural Methods for Programming: A Comprehensive Survey and Future Directions. *Appl. Sci.* **2025**, *15*, 12150. [CrossRef]
12. Sultan, B.; Apvrille, L. AI-Driven Consistency of SysML Diagrams. In Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems (MODELS 2024), Linz, Austria, 22–27 September 2024.
13. Yadav, P.S.; Rao, R.S.; Mishra, A.; Gupta, M. Machine Learning-Based Methods for Code Smell Detection: A Survey. *Appl. Sci.* **2024**, *14*, 6149. [CrossRef]
14. Eclipse. Capella. Available online: https://mbse-capella.org/ (accessed on 9 January 2026).
15. The REUSE Company. Systems Engineering Suite. Available online: https://www.reusecompany.com/systems-engineering-suite (accessed on 9 January 2026).
16. iRel40 Project. Available online: https://www.irel40.eu/ (accessed on 9 January 2026).
17. VALU3S Project. Available online: https://valu3s.eu/ (accessed on 9 January 2026).
18. de la Vara, J.L.; Morote, J.M. A Proposal for Model-Based Reliability-Oriented System Design in Industry. In Proceedings of the 21st IEEE International Conference on Software Quality, Reliability, and Security (QRS 2021), Hainan, China, 6–10 December 2021.
19. Morote, J.; de la Vara, J.L.; Giachetti, G.; Ayora, C.; Alonso, L. An Industrial Approach for Model-Based Reliability-Oriented System Design. In Proceedings of the 27th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2022), Beijing, China, 28 November–1 December 2022.
20. Eclipse. Capella Studio. Available online: https://github.com/eclipse-capella/capella-studio (accessed on 9 January 2026).
21. Lopez, B.; Álvarez-Rodríguez, J.M.; Parra, E.; de la Vara, J.L. Ontology Configuration Management for Knowledge-Centric Systems Engineering in Industry. In Proceedings of the 50th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2020), Valencia, Spain, 29 June–2 July 2020.
22. Rauzy, A.B.; Haskins, C. Foundations for model-based systems engineering and model-based safety assessment. *Syst. Eng.* **2019**, *22*, 146–155. [CrossRef]
23. Ticona Coaquira, F.J.; Wang, X.; Vidaurre Torrez, K.W.; Mamani Quiroga, M.J.; Silva Plata, M.A.; Luna Verdueta, G.A.; Murillo Quispe, S.E.; Auza Banegas, G.J.; Antezana Lopez, F.P.; Rojas, A. Model-Based Design and Testbed for CubeSat Attitude Determination and Control System with Magnetic Actuation. *Appl. Sci.* **2024**, *14*, 6065. [CrossRef]
24. Uludağ, Y.; Evin, E.; Gürbüz, N.G. Integration of systems design and risk management through model-based systems development. *Syst. Eng.* **2023**, *26*, 48–70. [CrossRef]
25. Valdivia-Dabringer, M.L.; Dybov, A.; Fresemann, C.; Stark, R. Towards Integrated Safety Analysis as Part of Traceable Model-Based Systems Engineering. *Proc. Des. Soc.* **2022**, *2*, 2005–2014. [CrossRef]
26. Zhang, Q.; Wang, S.; Liu, B. Approach for integrated modular avionics reconfiguration modelling and reliability analysis based on AADL. *IET Softw.* **2016**, *10*, 18–25. [CrossRef]
27. Rapin, N.; Bannour, B.; Adedjouma, M. Model-Based Generation and Analysis Toolset of Fault Trees with Heterogeneous Failure Events. In Proceedings of the 27th Pacific Rim International Symposium on Dependable Computing (PRDC 2022), Beijing, China, 28 November–1 December 2022.
28. Brusa, E. Digital Twin: Toward the Integration Between System Design and RAMS Assessment Through the Model-Based Systems Engineering. *IEEE Syst. J.* **2020**, *15*, 3549–3560. [CrossRef]
29. Vaicenavičius, J.; Wiklund, T.; Kavolis, D.; Draukšas, S.; Kalkauskas, A.; Vaicenavičius, R. SysIDE: SysML v2 textual editing and analysis system: Overview and applications. *CEAS Space J.* **2025**. [CrossRef]
30. Alanen, J.; Linnosmaa, J.; Malm, T.; Papakonstantinou, N.; Ahonen, T.; Heikkilä, E.; Tiusanen, R. Hybrid ontology for safety, security, and dependability risk assessments and Security Threat Analysis (STA) method for industrial control systems. *Reliab. Eng. Syst. Saf.* **2022**, *220*, 108270. [CrossRef]
31. Chen, R.; Chen, C.H.; Liu, Y.; Ye, X. Ontology-based requirement verification for complex systems. *Adv. Eng. Inform.* **2020**, *46*, 101148. [CrossRef]
32. Zhou, J.; Niemelä, E.; Evesti, A. Ontology-based software reliability modelling. In Proceedings of the Software and Services Variability Management Workshop—Concepts, Models, and Tools (SVM 2007), Helsinki, Finland, 24 April 2007.
33. Giachetti, G.; de la Vara, J.L.; Marín, B. Mastering Agile Practice Adoption through a Model-Driven Approach for the Combination of Development Methods. *Bus. Inf. Syst. Eng.* **2023**, *65*, 103–125. [CrossRef]
34. Delabeye, R.; Penas, O.; Plateaux, R. Scalable ontology-based V&V process for heterogeneous systems and applications. In Proceedings of the 14th System Analysis and Modelling Conference (SAM 2022), Montreal, QC, Canada, 23–28 October 2022.
35. Dong, M.; Wang, G.; Lu, J.; Wu, S.; Ma, J. Ontology formalism and semantic rules supporting traceability management in model-based systems engineering. *J. Eng. Des.* **2025**, 1–26. [CrossRef]
36. Jinzhi, L.; Zhaorui, Y.; Xiaochen, Z.; Jian, W.; Dimitris, K. Exploring the concept of Cognitive Digital Twin from model-based systems engineering perspective. *Int. J. Adv. Manuf. Technol.* **2022**, *121*, 5835–5854. [CrossRef]
37. Peugeot, T. GONG: An open source Ontology Based System Engineering toolset. In Proceedings of the 8th International Symposium on Systems Engineering (ISSE 2022), Vienna, Austria, 24–26 October 2022.

38. Maleki, E.; Fernandez, A.G.; Fischer, N.; Wijnands, Q.; Christofi, N. Semantic-based systems engineering for digitalization of space mission design. *Front. Ind. Eng.* **2024**, *2*, 1426074. [CrossRef]
39. Elaasar, M.; Rouquette, N.; Wagner, D.; Oakes, B.J.; Hamou-Lhadj, A.; Hamdaqa, M. openCAESAR: Balancing Agility and Rigor in Model-Based Systems Engineering. In Proceedings of the 15th System Analysis and Modelling Conference (SAM 2023), Västerås, Sweden, 1–6 October 2023.
40. Naouar, D.; El Hachem, J.; Voirin, J.L.; Foisil, J.; Kermarrec, Y. Towards the Integration of Cybersecurity Risk Assessment into Model-based Requirements Engineering. In Proceedings of the 29th IEEE International Requirements Engineering Conference (RE 2021), Notre Dame, IN, USA, 20–24 September 2021.
41. Voirin, J.L.; Bonnet, S.; Normand, V.; Exertier, D. Model-Driven IVV Management with Arcadia and Capella. In Proceedings of the 6th International Conference on Complex Systems Design & Management (CSDM 2015), Paris, France, 23–25 November 2015.
42. Sango, M.; Vallée, F.; Vié, A.C.; Voirin, J.L.; Leroux, X.; Normand, V. MBSE and MBSA with Capella and Safety Architect Tools. In Proceedings of the 7th International Conference on Complex Systems Design & Management (CSDM 2016), Paris, France, 13–14 December 2016.
43. Sango, M.; Godot, J.; Gonzalez, A.; Ruiz-Nolasco, R. Model-based system, safety and security co-engineering method and toolchain for medical devices design. In Proceedings of the 2019 Design of Medical Devices Conference (DMD 2019), Minneapolis, MN, USA, 15–18 April 2019.
44. Bitetti, L.; De Ferluc, R.; Mailland, D.; Gregoris, G.; Capogna, F. Model Based Approach for RAMS Analyses in the Space Domain with Capella Open-Source Tool. In Proceedings of the 6th International Symposium on Model Based Safety and Assessment (IMBSA 2019), Thessaloniki, Greece, 16–18 October 2019.
45. Brau, G.; Jenn, E.; Radu, S. A Capella-Based Tool for the Early Assessment of Nano/Micro Satellites Availability. In Proceedings of the 9th International Symposium on Model Based Safety and Assessment (IMBSA 2022), Munich, Germany, 5–7 September 2022.
46. Brunel, J.; Chemouil, D.; Rioux, L.; Bakkali, M.; Vallée, F. A Viewpoint-Based Approach for Formal Safety & Security Assessment of System Architectures. In Proceedings of the 11th Workshop on Model-Driven Engineering, Verification and Validation (MoDeVVA 2014), Valencia, Spain, 30 September 2014.
47. Ponnusamy, S.S.; Thebault, P.; Albert, V. Towards an Ontology-Driven Framework for Simulation Model Development. In Proceedings of the 8th European Congress on Embedded Real Time Software and Systems (ERTS 2016), Toulouse, France, 27–29 January 2016.
48. Hetherington, D.; Roques, P. STPA Analysis of Automotive Safety Using Arcadia and Capella. In Proceedings of the 11th European Congress on Embedded Real Time Systems (ERTS 2022), Toulouse, France, 1–2 June 2022.
49. Minacapilli, P.; Criado, F.; Campo, S.; Rodríguez, A.; Escudero, D. Small satellites mission design enhancement through MBSE and DDSE toolchain. In Proceedings of the Model Based Space Systems and Software Engineering (MBSE2022), Toulouse, France, 22–24 November 2022.
50. Anzen Engineering. Atica4Capella. Available online: https://www.anzenengineering.com/anzen-wiki/atica4capella/ (accessed on 9 January 2026).
51. Bogusch, R.; Ehrich, S.; Scherer, R.; Sorg, T.; Wöhler, R. A Lean Systems Engineering Approach for the Development of Safety-critical Avionic Systems. In Proceedings of the 8th European Congress on Embedded Real Time Software and Systems (ERTS 2016), Toulouse, France, 27–29 January 2016.
52. de la Vara, J.L.; Parra, E.; Ruiz, A.; Gallina, B. AMASS: A Large-Scale European Project to Improve the Assurance and Certification of Cyber-Physical Systems. In Proceedings of the 20th International Conference Product-Focused Software Process Improvement (PROFES 2019), Barcelona, Spain, 27–29 November 2019.
53. de la Vara, J.L.; García, A.; Valero, J.; Ayora, C. Model-Based Assurance Evidence Management for Safety-Critical Systems. *Softw. Syst. Model.* **2022**, *21*, 2329–2365. [CrossRef]
54. de la Vara, J.L.; Jiménez, G.; Mendieta, R.; Parra, E. Assessment of the Quality of Safety Cases: A Research Preview. In Proceedings of the 25th International Working Conference Requirements Engineering: Foundation for Software Quality (REFSQ 2019), Essen, Germany, 18–21 March 2019.
55. de la Vara, J.L.; Parra, E.; Alonso, L.; López, B.; Álvarez-Rodríguez, J.M. Integration of Tool Support for Assurance and Certification and for Knowledge-Centric Systems Engineering. In Proceedings of the 9th IEEE International Workshop on Software Certification (WoSoCer 2019), Berlin, Germany, 27–30 October 2019.
56. Parra, E.; de la Vara, J.L.; Alonso, L. Analysis of Requirements Quality Evolution. In Proceedings of the 40th International Conference on Software Engineering (ICSE 2018), Gothenburg, Sweden, 27 May–3 June 2018.
57. de la Vara, J.L.; Bahamonde, H.; Ayora, C. Assessment of the Quality of the Text of Safety Standards with Industrial Semantic Technologies. *Comput. Stand. Interfaces* **2024**, *88*, 103803. [CrossRef]
58. Mendieta, R.; de la Vara, J.L.; Llorens, J.; Álvarez-Rodríguez, J.M. Towards Effective SysML Model Reuse. In Proceedings of the 5th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2017), Porto, Portugal, 19–21 February 2017.

59. de la Vara, J.L.; Howard, P.; Ayora, C. An Ontology Metamodel for Knowledge-Centric Systems Engineering in Practice. In Proceedings of the 44th International Conference on Conceptual Modeling (ER 2025), Poitiers, France, 20–23 October 2025.
60. de la Vara, J.L.; Morote, J.M.; Ayora, C.; Giachetti, G.; Alonso, L.; Mendieta, R.; Muñoz, D.; Ruiz-Nolasco, R.; González, A. Early V&V in Knowledge-Centric Systems Engineering: Advances and Benefits in Practice. In Proceedings of the 18th IEEE International Conference on Software Testing, Verification and Validation (ICST 2025), Napoli, Italy, 31 March–4 April 2025.
61. Eclipse. Capella Days. Available online: https://mbse-capella.org/capella_days.html (accessed on 9 January 2026).
62. The REUSE Company. TRC Forum. Available online: https://www.reusecompany.com/trc-forums (accessed on 9 January 2026).
63. Transportation Safety Board of Canada. Swissair 111 Investigation Report—Executive Summary. Available online: https://www.tsb.gc.ca/eng/medias-media/fiches-facts/a98h0003/sum_a98h0003.html (accessed on 9 January 2026).
64. EASA. *Easy Access Rules for Acceptable Means of Compliance for Airworthiness of Products, Parts and Appliances (AMC-20)*; Amendment 19; European Union Aviation Safety Agency (EASA): Cologne, Germany, 2021.
65. Birolini, A. *Reliability Engineering: Theory and Practice*, 6th ed.; Springer: Berlin/Heidelberg, Germany, 2010.
66. Lazzaroni, M.; Cristaldi, L.; Peretto, L.; Rinaldi, P.; Catelan, M. *Reliability Engineering: Basic Concepts and Applications in ICT*; Springer: Berlin/Heidelberg, Germany, 2011.
67. O'Connor, P.D.T.; Kleyner, A.V. *Practical Reliability Engineering*; John Wiley & Sons: Hoboken, NJ, USA, 2012.
68. Stapelberg, R.F. *Handbook of Reliability, Availability, Maintainability and Safety in Engineering Design*; Springer: London, UK, 2009.
69. Antonino, P.O.; Trapp, M.; Barbosa, P.; Sousa, L. The Parameterized Safety Requirements Templates. In Proceedings of the 8th IEEE/ACM International Symposium on Software and Systems Traceability (SST 2015), Florence, Italy, 17 May 2015.
70. Fu, R.; Bao, X.; Zhao, T. Generic Safety Requirements Description Templates for the Embedded Software. In Proceedings of the 9th IEEE International Conference on Communication Software and Networks (ICSSN 2017), Guangzhou, China, 6–8 May 2017.
71. Westfall, L. *The Certified Software Quality Engineer Handbook*, 2nd ed.; ASQ Quality Press: Milwaukee, WI, USA, 2016.
72. Gallego, E.; Chalé-Góngora, H.-G.; Llorens, J.; Fuentes, J.; Álvarez, J.; Génova, G.; Fraga, A. Requirements Quality Analysis: A Successful Case Study in the Industry. In Proceedings of the Seventh International Conference on Complex Systems Design & Management (CSD&M 2016), Paris, France, 13–14 December 2016.
73. Runeson, P.; Höst, M.; Rainer, A.; Regnell, B. *Case Study Research in Software Engineering—Guidelines and Examples*; John Wiley & Sons: Hoboken, NJ, USA, 2012.
74. Roques, P. *Systems Architecture Modeling with the Arcadia Method: A Practical Guide to Capella*; ISTE Press: London, UK; Elsevier: Oxford, UK, 2017.
75. Lacrampe, S.; Thukral, V. Healthcare Use Case—Application of MBSE in managing patient care in patients suffering from malnutrition. In Proceedings of the 6th Annual Systems Engineering in Healthcare Conference (HWG 2020), Minneapolis, MN, USA, 28–30 April 2020.
76. Webel IT Australia. Examples of Applications of Systems Modelling Language (SysML) and Model-Based Systems Engineering (MBSE). Available online: https://www.webel.com.au/node/3517 (accessed on 9 January 2026).
77. The REUSE Company. From Zero to Hero: The Temperature War. Available online: https://www.reusecompany.com/webinars/from-zero-to-hero-the-temperature-war (accessed on 9 January 2026).
78. *ARP4754A*; Guidelines for Development of Civil Aircraft and Systems. SAE International: Warrendale, PA, USA, 2010.
79. *DO-254*; Design Assurance Guidance for Airborne Electronic Hardware. RTCA: Washington, DC, USA, 2000.
80. *IEC 62304*; Medical Device Software—Software Life Cycle Processes. IEC: Geneva, Switzerland, 2006.
81. *ISO 13485*; Medical Devices—Quality Management Systems—Requirements for Regulatory Purposes. ISO: Geneva, Switzerland, 2016.
82. *ISO 14971*; Medical Devices—Application of Risk Management to Medical Devices. 3rd ed. ISO: Geneva, Switzerland, 2019.
83. *EN 50126*; Railway Applications—The Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS). 2nd ed. CENELEC: Brussels, Belgium, 2017.
84. *EN 50129*; Railway Applications—Communication, Signalling and Processing Systems—Safety Related Electronic Systems for Signalling. 2nd ed. CENELEC: Brussels, Belgium, 2018.
85. *IEC 61508*; Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems. 2nd ed. IEC: Geneva, Switzerland, 2010.
86. Arduino. Arduino Mega 2560 Rev3. Available online: https://store.arduino.cc/products/arduino-mega-2560-rev3 (accessed on 9 January 2026).
87. Bourns. CAT 16. Available online: https://www.bourns.com/products/resistors/chip-resistor-arrays/product/CAT16 (accessed on 9 January 2026).