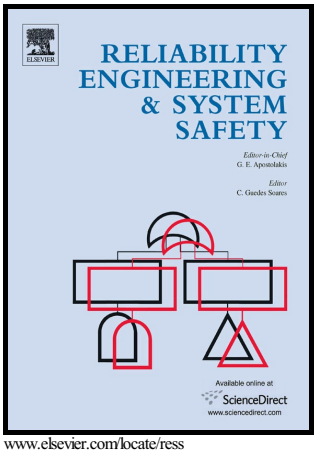
Safety Engineering with COTS components

Mark O'Halloran, Jon G. Hall, Lucia Rapanotti



www.elsevier.com/locate/ress

PII: S0951-8320(16)30829-8
DOI: http://dx.doi.org/10.1016/j.ress.2016.11.016
Reference: RESS5698

To appear in: *Reliability Engineering and System Safety*

Received date: 24 September 2013
Revised date: 19 February 2016
Accepted date: 17 November 2016

Cite this article as: Mark O'Halloran, Jon G. Hall and Lucia Rapanotti, Safet Engineering with COTS components, *Reliability Engineering and System Safety* http://dx.doi.org/10.1016/j.ress.2016.11.016

Safety Engineering with COTS components

Mark O'Halloran[a], Jon G. Hall[b,*], Lucia Rapanotti[b]

[a] *BAE Systems, UK*
[b] *Department of Computing and Communications, The Open University, UK*

Abstract

Safety-critical systems are becoming more widespread, complex and reliant on software. Increasingly they are engineered through *Commercial Off The Shelf* (COTS) (Commercial Off The Shelf) components to alleviate the spiralling costs and development time, often in the context of complex supply chains.

A parallel increased concern for safety has resulted in a variety of safety standards, with a growing consensus that a safety life cycle is needed which is fully integrated with the design and development life cycle, to ensure that safety has appropriate influence on the design decisions as system development progresses.

In this article we explore the application of an integrated approach to safety engineering in which assurance drives the engineering process. The paper reports on the outcome of a case study on a live industrial project with a view to evaluate: its suitability for application in a real-world safety engineering setting; its benefits and limitations in counteracting some of the difficulties of safety engineering with COTS components across supply chains; and, its effectiveness in generating evidence which can contribute directly to the construction of safety cases.

Keywords: Safety critical system, *Commercial Off The Shelf* (COTS) components, Problem orientation

1. Introduction

Safety-critical systems are defined as those systems where failure could result in the loss of life or injury to people, damage to equipment or to the environment [9]. Safety standards, e.g., [41, 17, 18, 6, 38], expect appropriate safety analysis tasks to be integrated within the design and development life cycle early, iteratively and on an ongoing basis. Generally this is taken to mean they should occur during the requirements capture and high level specification phases (e.g.,

*Corresponding author
Email addresses: mark.ohalloran@baesystems.com (Mark O'Halloran), Jon.Hall@open.ac.uk (Jon G. Hall), Lucia.Rapanotti@open.ac.uk (Lucia Rapanotti)

see [29, 39, 4, 10]), and is consistent with studies that have shown that a large proportion of anomalies occurs there [7, 22].

A challenging trend from an engineering viewpoint is the use of *Commercial Off The Shelf* (COTS) (Commercial Off The Shelf) components in safety-critical systems. COTS components ostensibly reduce costs and development time for complex systems. However, there are obvious difficulties in incorporating such 'black-box' components in the safety critical software engineering setting, and care is needed in the management of assumptions and expectations across supply chains.

Other work by the second and third authors has considered the challenge of early life cycle safety integration, leading to an approach to safety able to deliver early life-cycle models of requirements and high-level architectural design amenable to a wide range of safety analyses. This work is based on *Problem Oriented Engineering* (POE; see, for instance, [15, 14, 12]), an emerging framework for engineering as problem solving [13]. POE has developed (since 2001) into a collection of *thought tools* for the problem solving activities that underpinning design and engineering. As such POE research encompasses both theory and application, spanning the continuum from speculative thinking to experimentation and empirical work. POE has been extensively validated in industrial practice ([25, 26, 27, 32, 35, 28]) with increasing evidence of beneficial process improvement, which was also a concern in this study.

The work in this paper extends that effort in two directions. Firstly, it introduces and demonstrates how techniques for the explicit and incremental consideration of safety assurance can be used as a driver for design and development within the integrated approach, and can make a direct contribution to the related safety case in an efficient manner. Secondly, it examines the suitability of such techniques to meet the challenges of COTS-based system safety across supply chains, with particular focus on the establishment and communication of assumptions, expectations and safety requirements among multiple stakeholders and across organisational boundaries.

1.1. Paper structure

In Section 2 we review relevant background literature, with an overview of the Case Study in Section 3. In Section 4 we present the necessary underpinnings of Problem Oriented Engineering necessary to understand the approach. Sections 5 and 6 discuss, respectively, the case study and the safety case that was developed. We discuss and evaluate outcomes from the case study in Section 7 and conclude the paper in Section 8.

2. Background

2.1. Safety cases

A safety case [1] is a documented body of evidence providing a compelling, comprehensive and valid argument that a system is adequately safe for a given application in a given environment.

A safety case should address [41]:

- the management of risk commensurate with the potential risk posed by the system and its complexity;
- the validity of the safety requirements, i.e., that they are derived through thorough analysis and are traceable with respect to the system as designed and implemented, together with evidence of their satisfaction;
- and the well-foundedness of assumptions about the system, its operating environment or modes of use upon which the safety argument is based, with a justification that such assumptions are realistic and reasonable.

Standards also recommend that the safety case should contain not only evidence about the product, but also the process—attesting to good practice in development, maintenance and operation—and evidence on good engineering judgement and design.

Although the most convincing possible safety case is one which is formally valid and sound, Kelly and Weaver [21] observe that, due to the nature of the evidence in safety cases, a *provably* valid and sound case is unobtainable. In practice, there is wide-spread reliance on subjective expert judgement and claims of adherence to standards. This is recognised in current safety standards, which require only that the safety argument should be structured and evidence-based.

However, there remain acknowledged difficulties in the construction of safety cases:

- the combination of disparate pieces of the evidence, such as narrative, requirements, claims, plans, activities or goals is complex [2];
- traditionally, safety cases are developed after design and testing leading to the loss of *in situ* rationale for the safety aspects of the design and expensive re-design when the current design is indefensible [20], see Figure 1.

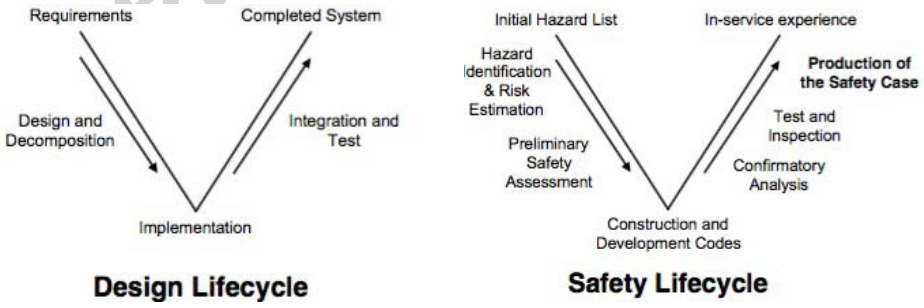


Figure 1: The traditional view of Safety Case Development (reproduced from [20]).

2.2. Commercial Off The Shelf Components

COTS components are commercial items sold in substantial[1] quantities in the commercial marketplace.

Since the mid 90s, there has been an increase in the use of COTS components in safety critical systems [40, 36], motivated by a need to reduce the cost and time pressure of development. Kelly *et al.* ([3]) and Lutz ([23]) both call for the development of better methods for COTS-based systems. Their call is, however, problematic with respect to certification given the black-box, i.e., hidden-state, nature of COTS components, and the fact that proprietary design information is typically limited.

Moreover, concomitant to the increase in the use of COTS components is a growth in complexity of the related supply chain. Menon and Kelly ([31]) observe that this poses particular challenges for the expression of safety requirements and their communication to multiple stakeholders across organisational barriers whose co-operation is necessary for the safe system development. They argue that many of these issues originate from unstated expectations or unjustified assumptions on the part of the varied stakeholders; exposing those assumptions during design is, therefore, desirable.

A feature of the case study discussed in the paper is the use of a variety of COTS hardware and software components, each carrying independent safety certification based on civil safety standards such as IEC 61508 and DO-178B, and supplied by different manufacturers and system providers in a complex supply chain.

2.3. The organisational context

At the time of the case study, the Integrated Platform Management System (IMPS) suppler organisation, to which the first author belonged, had adopted a development life cycle based on the V-model and in particular the framework outlined in the standard[2] for Defense System Software Development (DoD-Std-2167A), which followed a waterfall development process. Figure 2 gives an overview of this model: following the agreement of the contract requirements, further analysis was conducted to determine the hazards and risks before the initial specifications were generated and the interfaces determined; this was followed by a series of reviews against the design as it was developed. The development of a system concluded with a Test Readiness Review prior to conducting acceptance tests to validate the implemented solution. As shown, each of the phases provided outputs, in the form of evidence for the safety case.

While this process is robust and repeatable, the organisation continuously strives to operate more efficiently and effectively with fewer resources. Therefore, an overarching motivation for the work was to investigate how possible improvements could be achieved through the introduction of new practices, in this case the application of POE within the existing V-Model.

[1] Where substantial is market dependent.
[2] Now obsolete.

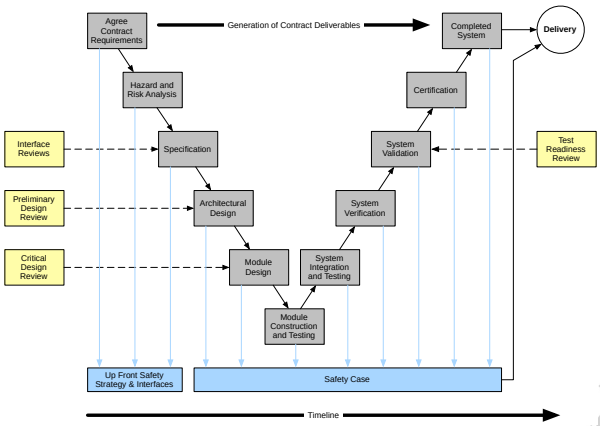


Figure 2: The IMPS supplier's adopted development model

3. Overview of the case study

3.1. The method

We take a case study approach [43] to investigate the early development life cycle of a safety critical distributed control system, an IMPS as used in naval vessels. A case study was deemed appropriate due to the need to investigate and evaluate the approach in the context of a real-world safety engineering process and in the natural organisational setting in which the process is exercised.

The specific safety critical system investigated was typical of those addressed by the first author's organisation through their engineering processes, and embodies specific features of interest such as the inclusion of COTS components across a supply chain. The first author has been involved with this type of system for the past 10 years as both a system provider and a customer and as such had first-hand experience of the key issues and difficulties associated with developing these complex systems.

Data were generated through the application of various POE techniques (see Section 4) throughout the exercise of the process; a subsequent qualitative analysis of the data was performed by the authors, where an interpretative approach [33] was taken. An evaluation of the study and our findings is included in Section 7.

3.2. The Purpose

The case study focuses on the problem of developing the communications interface between the control computers of the system (the *communications interface problem*). The IMPS is a safety critical distributed computing control system, providing control and monitoring functions for systems of a naval vessel, including propulsion, steering, electrical and auxiliary systems.

The IMPS for a given vessel will have a number of requirements (constraints, functional and non-functional) and in addition specific standards may be mandated. The development of an IMPS is generally outsourced by the shipbuilder to an IMPS supplier, who manages the development of the system prior to installation and then provides support prior to handing operation of the vessel and its systems to the end user.

For the IMPS under study, the contractual requirements mandated that Defence Standard 00-56 [41] should apply. From this a key requirement is that a safety case is prepared which provides compelling evidence that the system is tolerably safe, and commensurate with the approaches necessary for achieving the system's Safety Integrity Levels (SIL).

3.3. The Challenges

The functional requirements for the system were based on the system functions from previous vessels. In an effort to save costs, it was decided that rather than a bespoke solution, the system should utilise COTS hardware, such as Programmable Logic Controller (PLC)s, and associated software packages.

The COTS components used in this system carry independent safety certification based on civil safety standards such as IEC 61508 [17] and DO-178B [5]. As a consequence, the differences between these standards and Def-Stan 00-56 [41] required detailed consideration during the development phases.

One particularly problematic aspect of the IMPS was the interface between the central PLCs and the computers used to provide an Autopilot Control System (ACS): the PLCs have a number of proprietary communications protocols, none of which are available to the ACS. A need to develop a bespoke interface to provide safety critical communications between the systems arose, but it was unknown what the interface requirements would be and how they would be validated. This interface is the specific focus of the case study. Issues considered include: requirements elicitation and specification for the new technologies being used; their management across the supply chain; integration of legacy and new components and systems from diverse suppliers; safety verification and validation criteria for the components and the overall system.

3.4. System Overview

The hardware architecture (on the left of Figure 3) comprises two ACS computers, four Local Control Loop PLCs (LCL 1-4) and two Central PLCs (cPLCs). The ACS computers execute complex control algorithms necessary to maintain the vessel's positioning under numerous environmental conditions, operating as a master/slave pair of devices in order to provide hardware redundancy. LCLs 1-4 act on commands from the ACS computers to operate the control actuators of the vessel positioning system. The local control algorithm is closed-loop and continuously receives feedback from the actuator position sensors. The cPLCs interface with all of the vessel's sensors and provide this sensor data to the ACS. This data is sent to the Human Machine Interface (HMI) computer, which handles operator requests, such as new heading requests. The

computers and PLCs are interconnected via an Ethernet ring, with Ethernet switches located apporpriately in the network (not shown).

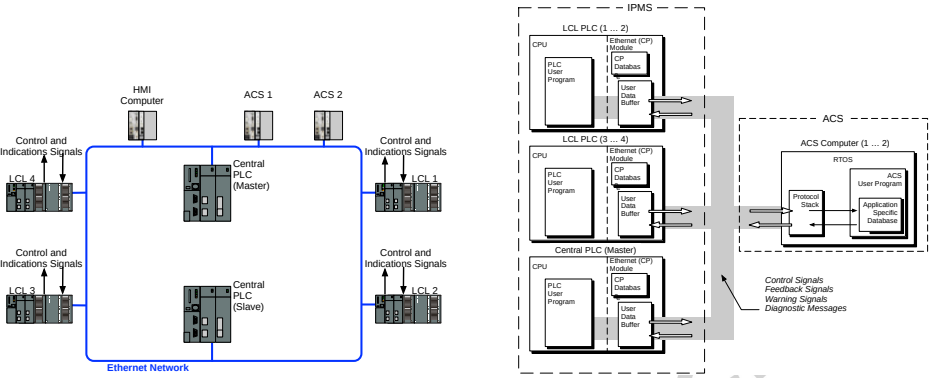


Figure 3: Hardware (left) and Software (right) architecture

The control algorithms of the ACS are bespoke programs and execute using a Real-Time Operating System (RTOS); the interface to the IMPS is via an Application Specific Database (ASD), which is used to store and manage the system parameters (input and output signals). The RTOS contains a protocol stack, which is compatible with a number of standard protocols, for instance TCP/IP and UDP/IP. The RTOS has also been certified for safety applications in accordance with DO-178B.

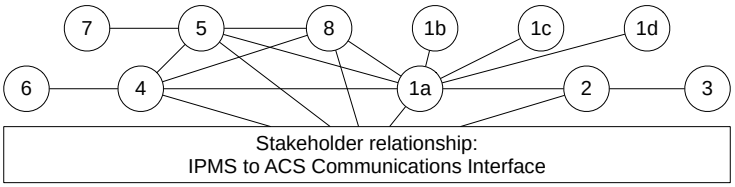
Both Local and Central PLCs are proprietary, developed to IEC 61131-3, executing proprietary programs, developed using their manufacturers' standards compliant software tools. This also applies to the interface blocks that implement the communications protocols. The PLC has an existing communications interface to the HMI Computer, based on ISO-TCP (RFC1006), which uses the PLC manufacturer's proprietary protocol, and has built-in diagnostics and so is not available in the ACS RTOS protocol stack.

Figure 3 (right) illustrates the PLCs and ACS computers connected on a common Ethernet bus. Each device has a network interface that handles the communications protocols before passing the data on to the main programs. For brevity we do not consider the design of the HMI to PLC interface in this study.

3.4.1. System Stakeholders

The ACS and IMPS were developed by different suppliers. A number of stakeholders had various claims on the system from a customer and regulator perspective (Figure 4 shows the relationship of stakeholders and lists their attributes). The Shipbuilder stakeholder (1) represents a group of internal stakeholders from Engineering (1a), Safety (1b), Supply Chain (1c) and Quality Assurance (1d). The Shipbuilder point-of-contact in this study was Engineering (1a) and in particular the delegated Design Authority for the IMPS. The set

of stakeholders associated with the shipbuilder was mirrored in the IMPS and ACS supplier organisations: i.e., for each supplier there was a Project Manager, Technical Authority, Safety Manager and Engineers.



Id	Stakeholder	Stake	Pwr	Urg	V.point	Role	Impact
1	Shipbuilder (a, b, c, d)	F & E	✓	✓	Customer	Active	High
2	MoD	F & E	✓	–	Client	Active	Medium
3	Royal Navy	E	–	–	End User	Passive	Low
4	IPMS Supplier	F & E	✓	✓	Developer	Active	High
5	ACS Supplier	F & E	✓	✓	Developer	Active	High
6	IMPS, PLC Approved Supplier	E	–	–	Supplier	Passive	Medium
7	ACT, RTOS Supplier	E	–	–	Supplier	Passive	Medium
8	Independent Safety Advisor	E	✓	–	Regulator	Active	Medium

Notes:

1. Stake: F – Financial: gain or loss to be made, E – Ethical: an interest in the system beyond economic
2. Pwr: Power – the ability to influence.
3. Urg: Urgency – the time sensitivity of the system development.

Figure 4: System stakeholders

3.4.2. System Requirements

The following initial requirements were elicited from suppliers.

#	Requirement	Description
1	Latency	The transit delay shall not cause the control system to become unstable, or delay warnings received at the operator positions.
2	Integrity	Adequate protection shall be used to ensure the data arrives in the format it was sent.
3	Reliability	Adequate protection shall be used to ensure data is not lost during transmission.
4	Confidentiality	Adequate protection shall be used to ensure data is not modified(!!!!!) during transmission.
5	Overhead	The overheads shall be kept to a minimum, this apples to processing time, memory usage and network bandwidth.
6	Resources	The development of the interfaces must be achievable against the available budgets and resources.
7	Support	The interface shall have the capacity to support design and growth margin retirements. As well as having a proven pedigree that should not suffer through component obsolescence.
8	Non-Functional Safety	The interface shall be commensurate with the safety standard Def Stan 00-56 and its requirements.

Additional high-level functional safety goals were expressed by the client (Ministry of Defence) to ensure that the vessel would remain safe under all operational conditions. These led to the following requirements with the aim to achieve and accurately maintain the vessel's ordered position, each under *normal*, *abnormal* and *emergency* operating conditions:

- To provide protection against the safe maximum roll angle;
- To provide an accurate measurement of vessels under keel depth for use by operators and control systems;
- To contribute to the automated control of the vessel's movement;
- To contribute to the maintenance of the watertight integrity of the vessel.

Finally, the following hazards were identified for the interface by the Shipbuilder and system suppliers, leading to the requirement that the system should provide sufficient mitigation agains those hazards.

#	Hazard Description
H1	Software error leads to an unplanned change in Control Mode, leading to a loss of vessel control.
H2	Software error leads to the autopilot causing spurious control actuator movement, causing potential collision or grounding.
H3	Spurious output from ACS resulting in the control actuator going to a 'hard-limit' position.
H4	No demand from ACS (to control actuators) when changing ordered position.
H5	Loss of feedback from the positional sensors of the control actuator, thereby resulting in loss of control of the vessel.

The study focuses on the application of various POE techniques to the communications interface between the PLCs of the IMPS and the computers of the ACS. A comprehensive introduction to POE can be found in [13]; in the next section we limit ourselves to a brief introduction of some of the basic concepts and notation relevant to the current case study.

4. Problem Oriented Engineering

POE is an emerging framework for engineering, the creative, iterative and often open-ended undertaking of designing and building products, systems and processes that address real-world problems. POE is *design theoretic* [13], by which we mean it provides a theory that characterises the elements of problem solving in terms of the effect they have on the process of design rather than on an artefact. Design in POE can use many types of design activities, including Weick's 'sensemaking' [42], various formal and informal refinement techniques, Jackson's problem progression [19], the use of architectures, etc, each of which is captured by the effect it has on design (see [13] for details).

Previous studies ([25, 26, 27]) have shown that POE[3] to be a good fit for system safety. In particular, *General Dynamics, UK* (GDUK) has used POE techniques in the safety-critical development of many military systems since 2007, including in the design of the stores management system for the Royal Navy Wildcat Helicopter and the Harrier JumpJet. GDUK has also used POE to introduce requirements models amenable to safety analysis in the early stages of their safety-critical product development process, thereby allowing early investigation of safety behaviour and identification of safety anomalies, and improving design processes for military systems. In the case of the Wildcat Helicopter, POE allowed the early identification of twelve interaction issues whose resolution led to an improved design.

One contribution to knowledge of that research is the *POE Safety Pattern* (PSP; [25]), shown on the right of Figure 6, a *process pattern*[4] for capturing high-level descriptions of system requirements and domain properties and assumptions through detailed problem models coupled with their traceable and justifiable step-wise transformation to specifications and high-level architectural design artefacts, with the essential quality that those problem models are amenable to various forms of safety analysis. The steps of the PSP are described in Table 1. The use of the PSP will be discussed in detail in the case study.

POE specialises Rogers' definition of engineering [37] to systems engineering as:

> "Systems engineering refers to the practice of organising the design and construction of any system which transforms the physical

[3] Or, rather, POSE Problem Oriented Software Engineering, [14], a specialised theory for software engineering that is embedded within POE.

[4] I.e., a pattern through which appropriate processes can be instantiated.

world around us to meet some recognised need."

As such, systems engineering becomes a problem solving exercise, the problem being, given a physical environment E, to find the system S that meets a real-world need N to the satisfaction of a group of stakeholders K, written $E(S)$ $\mathbf{meets}_K$ N.

Each of E, S and N are typically complex objects: E (resp. S) being formed from a collection of *domains* (resp. components), with N being, perhaps, a collection of use cases, user stories, requirements clauses, etc. We thus use a number of notations, graphical and otherwise, to represent and illustrate problems, from natural language, causal calculi, program code, to a problem diagram-like notation [19] (see Figure 5).

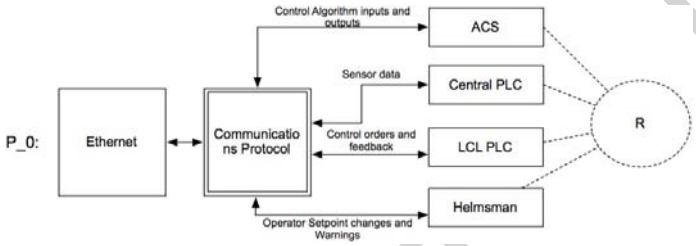


Figure 5: A problem diagram-like representation of the system of Figure 3. See Table 8 for a description of the elements of this problem diagram.

A design is a sequence of *solvability preserving transformations* that move a problem to known solved problems. Problem transformations relate a *conclusion problem* P to a collection of *premise problems*, P_i, $i = 1, ..., n$, $(n \geq 0)$, via a *step rationale* J .

$$\frac{P_1 \quad ... \quad P_n}{P} \langle\langle J \rangle\rangle \tag{1}$$

By identifying premise and conclusion problems, such transformations build into design trees. Figure 7 shows the whole design tree for the case study, to be explicated in the sequel.

During design, POE interleaves analysis in and of the problem space with synthesis in and of the solution space:

- in the problem space, the problem is understood and agreed with validating problem stakeholders, such as the customer, domain experts, etc;
- in the solution space, a design for the solution is created and evaluated with validating solution stakeholders, such as the regulator, end users, etc.

Within the POE 'toolkit', the PSP is a form of *Assurance-Driven Design* (ADD; [11]) through which assurance is seen as a driving force in the design of a system rather than as a 'bolt-on'. ADD results from the interpretation of Equation 1 not as a relation between a conclusion problem and a set of premise problems mediated by a step rationale, but as a relationship between

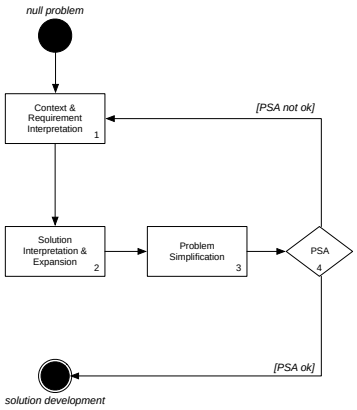


Figure 6: The POE Safety Pattern

a (conclusion, step rationale) pair and a set of premises. This places the step rationale, and so the safety case which will be derived from it, on a par with the solution artefact: any step towards a solution must consider both assurance and product needs. Pressing assurance concerns, discovered during the exploration of J_0, are then allowed to drive problem solving.

5. Case study

Given the emerging importance of COTS within safety-critical system development and that little is known about the construction of safety cases that involve them, we speculatively investigated the application of ADD (through the PSP) to evaluate its benefits and limitations. The full development can be read in the technical report that accompanies this paper [34]. Here, we give some highlights of the development that form the basis of the evaluation. The reader may wish to refer to Figure 7 throughout this section, together with accompanying domain descriptions (Table 8), phenomena descriptions (Table 9) and requirements (Table 10).

All development in POE begins with the technical device that is the *null* problem, P_{null}. The *null* problem represents the existence of a problem of which no detail is known and forms the root of a development tree, such as that in Figure 7. As mentioned above, ADD considers a problem, step rationale pair and so, alongside, P_{null}, we must consider the initial step rationale J_0. This motivates *problem exploration*, by which the problem is populated, as are any associated *concerns*.

5.1. Stage 1: Problem exploration

During problem exploration, the real-world domain and system requirements were discovered, based on information gained through safety assessments and design reviews for previous vessels, as well as meetings and correspondence among

Table 1: PSP safety steps

Stage	Aim	Applicable techniques	Artefacts generated	Typical validating stakeholders
Problem Exploration	To build problem models by capturing knowledge of requirements and relevant context properties	POE Requirement and Context Interpretations	Problems, Transformations and Justifications	Customer; Domain experts
Architectural Exploration	To embed high-level architectural design in a problem model	POE Solution Interpretation and Expansion	AStructs [12], Problems, Transformation and Justifications	Project Safety Engineer; Independent Safety Expert
Problem Refactoring	To ensure readiness of problem models for subsequent safety analysis	POE Problem Progression; various POE Interpretations	Problems, Transformations and Justifications	Project Safety Engineer; Independent Safety Expert
Preliminary Safety Analysis	To apply standard safety analysis techniques to problem models	standard safety analysis techniques (e.g. HAZOP, FTA, simulation)	Problems; standard artefacts produced by the applied safety techniques	Independent Safety Expert

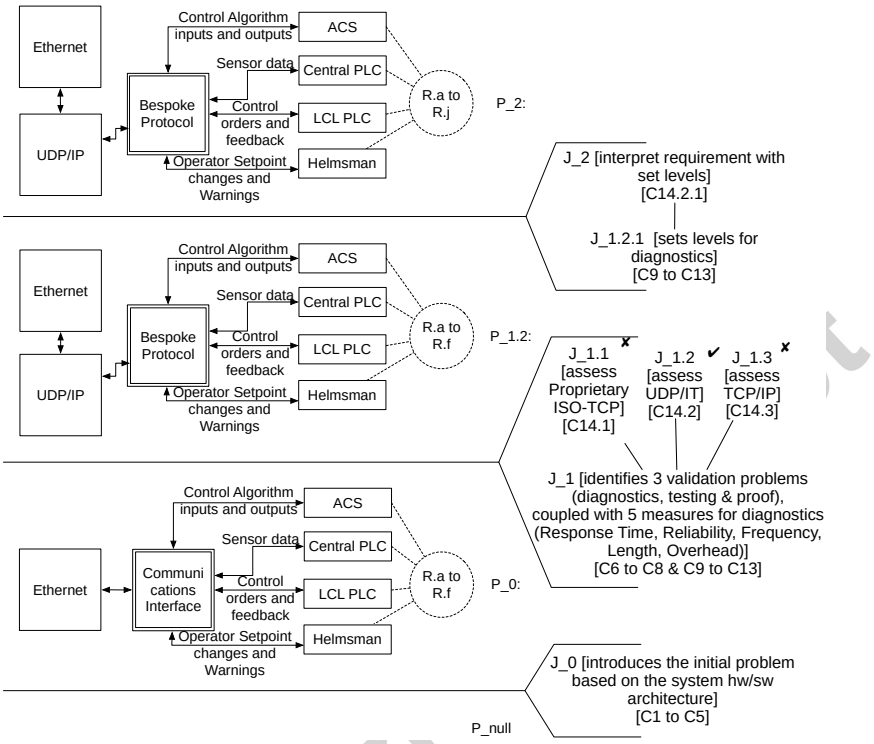


Figure 7: Development tree, from the initial 'null' problem. Development is upwards.

stakeholders for the IMPS under development. The outcome, P_0 (see Figure 7), reflects what was understood at that point in the process about the architecture and the basic requirements for the communications interface. In particular, it identifies that the context of the development, including all PLCs, and the target of the problem solving exercise is the design of the Communications Interface.

As part of the exploration of J_0, then, we must identify the concerns that arise during problem exploration, the discharge of which will justify the step rationale.

In the case study, five concerns[5] are discovered and must be discharged before the final assurance case can be said to be complete. The specific nature of their discharge is out of scope here (but see [34] for full details): it may be based on analysis, expert review, *etc*, as in the case of three of the identified concerns, or may require further exploration of the problem or solution as is the case, for instance, with the discharge of hazards. They are:

[5]Including various standard concerns; see [24] for details.

Domain	*Description*
Autopilot Control System (ACS)	The computer responsible for issuing control orders based on an operator requested set-point, feedback position indications and the output of a control algorithm. This is a COTS computer and runs a proprietary RTOS. Its application program is to be developed by the ACS supplier.
Central Programmable Logic Controller (Central PLC)	The Central PLC responsible for data acquisition and distribution from the operator and field inputs. This is a COTS PLC and runs a proprietary operating system. The PLC program is to be developed by the IMPS supplier.
Local Controller (LCL PLC)	A local PLC responsible for providing local actuation of control valves based on demands of the Navigation Control System. This is a COTS PLC and runs a proprietary operating system. The PLC program is to be developed by the IMPS supplier.
Ethernet	Communications medium between all the computers.
Helmsman	The operating position responsible for distributing requests for changes in course and also monitoring the vessel's position. (The helmsman interacts with the system through the HMI Computer, a connection domain abstracted away in this analysis.)
Communications Interface	The interface to be designed

Figure 8: Domains descriptions for Figure 7.

Phenomenon	*Description*
Control orders	New demand signals
Feedback	Feedback position signal from the control actuator
Setpoint changes	Operator entered requests for a new vessel position
Warnings	Warnings alerting the helmsman that a demand has failed
Sensor Data	Signals for position and speed of the vessel

Figure 9: Phenomena descriptions for Figure 7.

Requirement	*Description*
R.a	The ACS shall transmit orders to the LCL PLCs.
R.b	The LCL PLCs shall provide actuator feedback information to the ACS.
R.c	The Central PLC shall provide sensor data to the ACS.
R.d	The ACS shall send warning information to the Central PLC for display at the HMI.
R.e	The HMI shall transmit Operator set-point changes to the ACS via the Central PLC.
R.f	Network diagnostic information shall be made available to the ACS and Central PLC.

Figure 10: Requirements descriptions for Figure 7.

STEP FROM P_{null} TO P_0: *Design Space*

JUSTIFICATION J_0: The problem model of Figure 5, including the problem diagram and its associated descriptions, is the result of an evaluation of the real-world environment and the system requirements, informed by safety assessments and design reviews for previous vessels, and by meetings and correspondence among stakeholders of the system under development. POE context interpretation classed each device as a domain, while POE requirement interpretation led to the stated requirements R.a to R.f.

CONCERN: C1: Interpretation Concern
STATUS: Discharged

CLAIM: The interpretations are well-founded

ARGUMENT & EVIDENCE: The choice of domains follows from the vessel functions, legacy system architecture and components that are being replaced by COTS components, and the need for a communications medium (Ethernet) enabling the Communications Interface.

CONCERN: C2: Hazard Identification Concern
STATUS: Pending

CLAIM: Sufficient hazard identification has been conducted

ARGUMENT & EVIDENCE: Hazards have been determined by the Shipbuilder and system suppliers (H1 to H5) and are captured in their Hazard Log. However, this concern cannot be discharged until the solution has been assessed.

RISKS: All hazards are not identified

CONCERN: C3: Hardware Reliability Concern
STATUS: Discharged

CLAIM: Hardware reliability is commensurate with the target safety level

ARGUMENT & EVIDENCE: The IMPS supplier has developed Reliability Block Diagrams for the hardware architecture in the scope of their supply. The results of this analysis show that the system has a Probability of a Dangerous Failure in excess of the requirement for a SIL 2 system.

CONCERN: C4: Network Topology Concern
STATUS: Discharged

CLAIM: Network Topology is robust enough to support the communications interface

ARGUMENT & EVIDENCE: The IMPS supplier has held and passed a preliminary design review to assess the requirement of the network, which established that a standard Ethernet network shall be provided. The individual hardware components have been assessed at design reviews and have undergone factory acceptance testing as standalone items. The hardware has subsequently been delivered to the shipbuilder for integration into the vessel. An independent assessment

of the network was conducted by two independent advisors; their recommendations have been implemented where practicable. The shipbuilder has conducted a vulnerability assessment on the network components outside the scope of supply of the IMPS supplier: this resulted in a need to re- route the network to avoid critical failures in the event of loss of power.

CONCERN: C5: Communications Interface Concern
STATUS: Pending

CLAIM: The communications interface meets the requirements of the system

RISKS: The solution does not provide the necessary Quality of Service. The solution cannot be implemented and tested with the available resources. Evidence cannot be generated to support the safety case. The implemented interface does not mitigate against the identified hazards (H1 to H5).

The two pending concerns, C2 and C5, trigger the following exploration steps to establish a candidate solution software architecture driven by its validation criteria.

5.2. Stage 2: Solution exploration

This step consists of an assurance-driven exploration of the solution to be designed and its architecture, which results in problem $P_{1.2}$ and justification J_1, with one component for each choice: $J_{1.1}$, $J_{1.2}$, and $J_{1.3}$. Substantial effort in this step consists of work in the validation space to form and analyse the justifications. Full details are contained in [34]; here we describe only the most salient features.

5.2.1. Diagnostics, testing and proof

The communications interface development was to be conducted in accordance with the processes and techniques required to achieve SIL 2. Therefore, the same level of evidence was required in terms of the requirements, design and test documents, and this evidence needed formally documenting by the suppliers.

To implement a fit-for-purpose bespoke solution and provide safety justification (hence addressing concern C5) the stakeholders agreed that appropriate levels of diagnostics, testing and proof should be mutually agreed between the shipbuilder, the IMPS supplier and the ACS supplier.

Diagnostics were required to ensure the reliability, integrity and security of the communications interface, as the Quality of Service of the communications can cause control system instability

The level of testing had to be sufficient to demonstrate that the communications interface was robust enough for the volumes and types of data transmitted and received across the entire IMPS network.

This is captured in the step rationale in which new concerns were derived from C5:

C6 Level of Diagnostic: The level of diagnostics is sufficient to ensure the reliability, integrity and security of the data. Status: Pending

C7 Level of Testing: The level of testing is sufficient to ensure the quality, safety and performance of the selected protocol. Status: Pending

C8 Level of Proof: The level of proof is sufficient to meet the safety requirements of the system. Status: Pending.

C9 Response Time: The allowable time for a message to take to transmit from one device to another is set so not to affect the stability of the controllers. Status: Pending

C10 Reliability: The allowable number of lost messages during transmission of data is set so not to affect the stability of the controllers. Status: Pending

We note that, as might be expected, all concerns in this step are pending the solution; the nature of ADD encourages the discharge of these concerns to drive the development of that solution: for instance in the case of C8, we might anticipate that SIL 2 level of evidence will be required from the suppliers, and this establishes key constraints by which the solutions delivered will be validated. Without knowledge of the concerns, such constraints and other requirements may be lost during development.

5.2.2. Architectural interpretations and expansion

Having established key validation criteria for the solution, three candidate architectures for the communications interface were assessed. The outcome is that a bespoke common communications protocol is to adopted and developed by both suppliers, with the choice being between a proprietary ISO-TCP protocol, versus one developed over UDP/IP or TCP/IP.

Step justification J_1 was developed, pending concerns for which led to the architecture shown in Figure 11, based on the standard UDP protocol. For more details of the choice, see [34].

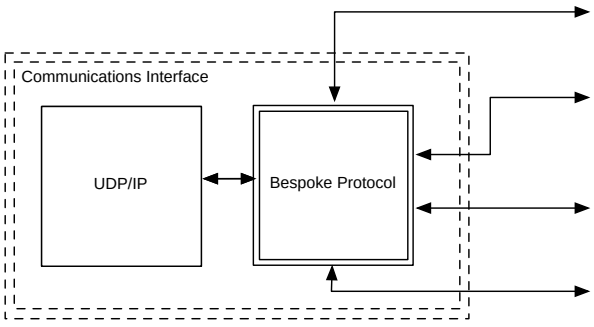


Figure 11: The chosen UDP-based candidate architecture

This step introduced one further concern:

C13.2 Solution Feasibility: The solution can be implemented by both suppliers and provides a suitable protocol in terms of the identified requirements. Discharged, due to the use of the standard UDP protocol.

5.3. Stage 3: Problem Refactoring

Problem refactoring ensures PSA readiness, i.e., to make sure that the problem model is in a form which allows the necessary PSA to be conducted. In this case study, this entailed some validation work related to establishing values for the diagnostic parameters which are the subjects of concerns C9–C12, coupled with a re-interpretation of the requirements to bring such values into the design space. (Examples of more complex transformations are given in safety critical developments reported in [24]).

This led to concerns C9: Response time, C10: Reliability, C11: Frequency, and C12: Length, all of which were discharged through the estimation of network parameters through simulation, done as part of the justification step.

5.3.1. Requirements interpretation

With the validation parameters set, the requirements were then interpreted to bring them into the design space, adding R.g through R.j as follows, and leading to problem P_2 in Figure 7, the last we consider here:

R.g Message size shall be limited to 240 bytes (constraint of the PLC communication block).

R.h Maximum system latency shall not exceed 800ms (this is a combined latency for the feed-forward and feedback paths).

R.i Maximum packet loss shall not exceed 5 packets.

R.j Message frequency shall be set to 5Hz (200ms).

and the solution feasibility concern (C13.2) was refined accordingly, to

C13.2.1 Solution Feasibility: The requirements specified shall enable the development of a robust, reliable and secure interface that supports the control system.

which remains pending.

5.4. Stage 4: Preliminary Safety Analysis

As already mentioned, the aim of Preliminary Safety Analysis (PSA) is to:

- confirm any relevant hazards from the system level hazard list;
- identify if further hazards need adding to the list; and
- analyse the chosen architecture to validate that it can satisfy the safety targets associated with the identified relevant hazards.

A number of techniques can be applied to perform a PSA: for instance, Mannering [24] applies both FTA and HAZOP.

We have already seen how applicable hazards H1 to H5 were identified from previous vessels (see Section 3.4.2). In order to identify further possible hazards and gain stakeholder approval, a HAZOP study was conducted (see, for instance, [30]). The selection of HAZOP guidewords and the choice of material to be used in the case study was part of the HAZOP preparation and was based on the understanding of the interface gained during the application of PSP.

For brevity and given it is a standard technique, we provide only a brief summary of the PSA HAZOP. The HAZOP was conducted as a workshop with the key stakeholders, with relevant supporting documents provided to the participants prior to the workshop. The HAZOP was based on the problem characterisation given in P_2. The objective was to assess the hazards associated with any failures and their impact on the system and its environment. The outcome of that PSA HAZOP was that no further hazards were identified beyond H1 to H5.

The PSA also aided the assessment of concerns C9 to C12, associated with the set levels of diagnostics, and made a number of recommendations for safeguards to be implemented. From this assessment, further requirements for the diagnostics were derived and included in the interface specification to undergo further reviews by the stakeholders. The safeguards are captured in the following steps:

STEP PRELIMINARY SAFETY ANALYSIS*: Validation space*

JUSTIFICATION $J_{2.1}$: The PSA assessed concerns C9 to C12 and made a number of recommendations for safeguards to be implemented. Here is the full step justification:

CONCERN: C14: Safeguards
STATUS: Pending

CLAIM: Appropriate safeguards are implemented

ARGUMENT & EVIDENCE: It is expected that the following safeguards will be implemented in the application and session layers of the communications interface:

- Monitor the health of the receiving and transmitting nodes, using a communications heartbeat signal between the nodes.
- Send next order after predetermined period (set to 200ms).
- Generate an operator warning when a node is not communicating (Node Down Warning).
- Apply a timestamp to each message, and reject old messages.
- Generate an operator warning when the measured value deviates from the ordered value (Set-point Error Warning).
- Implement a limited number of message resends to a node based on the frequency and lost packet requirements.
- Apply a Cyclic Redundancy Check (CRC) to each message.
- HMI should receive feedback from ACS before displaying new values.

- Perform signal comparisons between redundant hardware and raise warnings if differences are detected (Signal Deviation Warning).

6. Safety case

Based on the original SIL assessments for the case study, Def-Stan 00-56 places the IMPS and ACS as 'Medium' integrity systems. For this level of integrity of the communications interface, the evidence should be provided as a safety case and be able to show that:

- safety requirements are defined, correct and sufficient to maintain the safety of the system;
- hazards have been identified and assessed and risk reduction carried out;
- the domains external to the interface in which data is manipulated has been subject to hazard analysis;
- hazards that could adversely affect the integrity of the interface have been controlled via the definition of safety requirements for the operational processes that manipulate data.

The raw evidence generated throughout the development was extensive, including large amounts of test evidence, review evidence, quality assurance evidence, as well as the deliverable items, much of which is contained in [34].

The overarching safety justification for the IMPS (covering both hardware and software) was a product- and process-based argument in order to meet the requirements of Def-Stan 00-56 that the IMPS was tolerably safe. Def-Stan 00-56 defines tolerable as a level of risk between broadly acceptable and unacceptable that may be tolerated when it has been demonstrated to be As Low as Reasonably Practicable (ALARP). For the IMPS this meant that all individual hazards were documented and the associated risks were demonstrated to be acceptable, and a robust argument provided that the IMPS was justified against the Safety Roles identified in Section 3.4.2. As a consequence a GSN argument was constructed, following a top-down approach, to argue that the IMPS was tolerably safe.

Note that the concerns raised during the PSP process identified three main areas that needed to be addressed. The first concern, associated with the level of diagnostics for the interface, was progressed through to conclusion by recursively applying PSP as indicated in the previous section, and resulted in a set of requirements that allowed development work to commence. The second and third requirements concerned the level of testing of the system, hence were not addressed by PSP, but instead provided a focus for integrating PSP work into the wider safety justifications for the interconnected systems, and as such aided the development of their respective safety cases: even though exhaustive testing of the interface will not be possible, an approach was planned to provide

sufficient test coverage to discharge the stakeholder concerns, captured through the inclusion of GSN goals for an agreed set of test plans and procedures. This allowed the suppliers to test their standalone systems prior to integrating the systems and conducting the acceptance tests using IMPS test environment. Furthermore, the tests to discharge the final stakeholder concerns were focused on the hazards and risks identified during the PSP activities.

As the communications interface was developed by two separate organisations with differing software processes and tools, a modular approach to the safety case was deemed appropriate. The modular approach related the communications interface safety argument to the safety arguments for both the ACS and IMPS, as they all contribute to the overall Vessel Safety Case. The modular approach to creating the safety argument began with creating modules in the safety case for each of the software modules in the software architecture, then defining the top-level argument that is to be justified. The modular top-level argument for the case study is shown in Figure 12. The 'IFSafe' module is the contract module (based on the GSN extensions discussed by [8]), and a Safety Case Contract was used to record the dependencies that existed between the safety argument modules (within the figure, for brevity, only the contract between the IMPSSafe and IFSafe modules is shown). The organisation responsible for overseeing these contracts was the shipbuilder.

Further detail of the safety case is given in [34].

7. Discussion and evaluation

In this study, we have investigated how the assurance needs of a safety-critical development can use to drive the early life cycle of a live project for the development of an IPSM used on naval vessels. The system was a safety-critical distributed control system, which was developed in the context of the normal engineering practice of the IPSM supplier organisation. A particular challenge on this project, which was the focus of the case study, was the development of the communications interface, due to the need to integrate legacy and new components, making use of COTS components across a supply chain. The four steps of the PSP were applied iteratively in the context of the V-model process adopted by the IPSM supplier organisation.

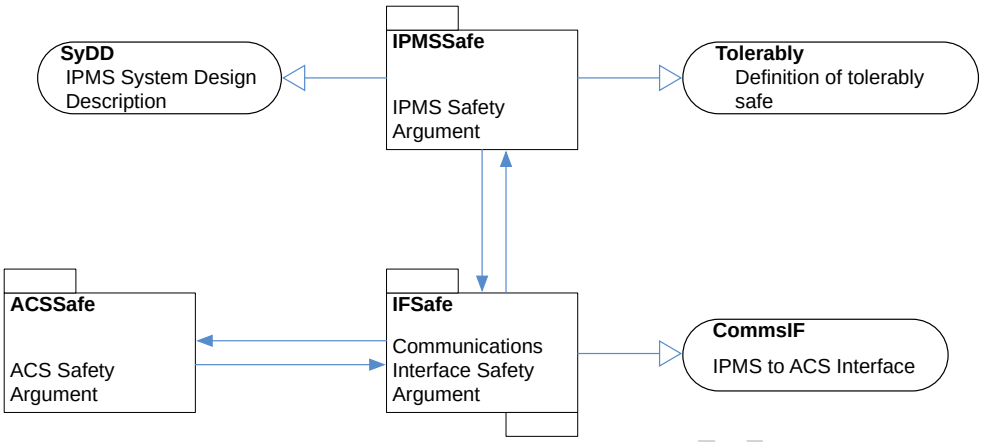
In this section we discuss the outcome of the case study in view of the work's set objectives.

7.1. Suitability for systems engineering practice

The case study has demonstrated that applying ADD resulted in a verified solution, which links to the V&V activities associated with the V-model, in the normal exercise of the process in an industrial context.

It was shown that the approach had the ability to:

- engage various stakeholders and capture their participation throughout the analysis, specification and design stages related to the V-model;



Argument Module:		Top-Level Argument		
		Identifier	**Summary**	**Module**
Goal Addressed		IPMSSafe	IPMS is tolerably safe	IPMS_Argument
Solution Presented		None		
Context Defined		SyDD	IPMS System Design Description	
		Tolerably	Definition of tolerably safe.	
Goals Requiring Support		None		
Inter-module Dependencies	**Goals**	IFSafe	Communications Interface is tolerably safe.	IF_Argument
		ACSSafe	ACS is tolerably safe.	ACS_Argument
	Solutions	None		
	Context	CommsIF	The interface is between the IPMS and the ACS.	

SAFETY CASE MODULE CONTRACT

Participant Modules

IPMSSafe; ACSSafe; IFSafe

Goals Matched Between Participant Modules

Goal	**Required by**	**Addressed by**	**Goal**
G1.12	*IPMSSafe*	*AccTestArg*	*G2.113*
G1.2111	*IPMSSafe*	*IFSafe*	*G2.21*
G1.213	*IPMSSafe*	*IFSafe*	*G2.21*

Collective Context and Evidence (Solutions) of Participant Modules held to be consistent

Context	**Evidence**
C5 (RefSet)	*Sn2.113*
C1.15 (PSP)	*Sn1.2131; Sn2.21*

Resolved Away Goal, Context and Solution References between Participant Modules

Cross Referenced Item	**Source Module**	**Sink Module**
Sn1.12	*IFSafe*	*IPMSSafe*
Sn1.2131	*IFSafe*	*IPMSsafe*

Figure 12: Top Level Argument with Safety Contract

- capture the stakeholder requirements and provide rich traceability [16] of the solution in the form of transformation diagrams and structured prose;
- lead, through stakeholder participation, to a validated solution with an agreed safety justification.

Therefore, the outcome of this case study contributes to the growing body of evidence concerning the ability of the approach to combine effectively with safety practice.

7.2. Working within supply chains

The ability of the approach to engage various stakeholders and capture their participation throughout the analysis, specification and design stages of the process was considered a key success factor in this case study, where the complexity of the engineering task was compounded by the need to communicate knowledge and reach a common understanding among diverse stakeholders across the supply chain.

PSP data was collected with the aid of stakeholders from different organisations, and, while misinterpretations are always possible and a total guarantee that the specification is error free is unachievable, the involvement of the stakeholders in validating the PSP justifications, and in conducting the PSA provided a high level of assurance as to the soundness of the data.

The approach was effective in identifying, expressing and sharing appropriate safety requirements across the supply chain, making explicit and justifying key assumptions, and allowing for appropriate consideration of the various certification standards and the generation of supporting evidence for the safety case. Therefore, it demonstrated that the approach can contribute directly to mitigating the challenges acknowledged by Menon and Kelly [31], by avoiding unstated expectations or unjustified assumptions on the part of the varied stakeholders.

7.3. Contribution to safety cases

The PSP and GSN were conceived to perform different functions: the PSP to support software development for safety-critical applications by exploiting POE techniques to solve engineering problems, while GSN to develop safety arguments. Also, the PSP activities are fully integrated within the chosen development process, while GSN allows one to create the safety arguments alongside, but separately from, the chosen software development method.

Each approach can benefit the development process. The PSP engenders stakeholder participation, promotes the continuous validation of problem and solution artefacts, allows for structured prose to record critical parts of justifications; also, hazard and risk identification is inherent to the process, and design trees and transformation diagrams provide a traceable development, from problem to solution. However, the PSP is not widely known or used by the safety community.

On the other hand, GSN allows for a structured hierarchical breakdown of the safety case which captures the most important aspects of safety arguments,

can be used during various stages of argument development, is semantically well defined and understood — a standard has been drafted, and is increasingly being adopted by organisations in various industries. However, GSN is difficult to write (but easy to read), does not prevent bad arguments being created, and can be subjective due to the lack of stakeholder endorsement throughout the process.

As a consequence, in the case study, we chose to combine the two with a view to capitalise on their relative strengths, with the outcome that the PSP has allowed for stakeholder participation culminating in a validated solution with an agreed safety justification, with GSN providing a clear overarching structure for the safety argument within which evidence from the PSP could be integrated alongside other forms of evidence, and providing a route to address the stakeholder concerns associated with levels of evidence and testing. The result of this integration was a product and process based safety argument for the communications interface, which in turn created a safety contract for the suppliers and shipbuilder, thus ensuring each organisation's obligations are fulfilled.

Note that there are some limitations and sparse information on how to construct these contracts in a tabular form, as discussed by [8] and also experienced in this case study. That said, for the purpose of managing the interface between the two suppliers this approach did have its benefits, especially when ensuring that each supplier was aware of their obligations.

7.4. Threats to Validity

Above we claim that our approach is engaging for stakeholders, offers rich traceability, led to validated solutions and aided in communications. External validity concerns the problem of knowing whether a case study is generalisable. The findings we have presented argue, from a single case study and, unfortunately, the opportunities for doing so within the same sector are few and far between: our research depends on the availability of a practitioner/domain expert with problems to solve and a willingness to explore new approaches to doing so. However, as Yin counsels [43], case studies rely on analytic generalisation, where the investigator is 'striving to generalize a particular set of results to some broader [context].' and in that broader context, we find that our approach has these characteristics is a repeated conclusion of other case studies that we have conducted both with the safety-critical systems industry and in other sectors, for instance, [24, 32, 28]. Given the generality of the approach and its domain independence, we have some small confidence that these general conclusions will hold broadly across engineering disciplines.

Having said that, however, this case study can be seen as reapplying the work of Mannering ([24]) which argues that that work provides an operational basis for conducting research in this area. I.e., given the availability of similar case studies, we have some confidence that the experimental procedure followed here is repeatable.

8. Conclusion

This study has successfully applied the PSP in the context of industrial practice and contributed to a growing body of evidence of the benefits of doing so. However, given the limited scope and timescale of the case study, the results, whilst encouraging, remain preliminary and fall short from introducing a paradigm shift in how safety software and cases are produced. Further research research is needed, particularly in the following areas.

Further applications of the approach in within safety practice would be beneficial, both to problems of growing complexity and in the context of organisations adopting diverse development approaches. Such research would produce further evidence of the applicability and effectiveness of the approach, could be used to draw comparisons and further validate the findings of this investigation, and could further promote the use of the PSP within the software safety community.

Besides GSN, alternative methods for creating safety arguments exist; further studies could compare and contrast these methods and investigate to which extent the PSP is compatible with them. This could also be further expanded to look for patterns for the re-use of safety arguments, with the aim of making the safety case development process more efficient.

Another possible line of enquiry concerns the development of a PSP standard, which provides clear guidelines on how to conduct the activities and also provides guidance on presenting the outcomes in the form of a safety case, with or without reliance on other methods (such as GSN). These could also be accompanied by software tool support or drawing templates for use in existing software packages.

References

[1] Bloomfield, R., Bishop, P., Jones, C., Froome, P., 1998. ASCAD - Adelard Safety Case Development Manual.

[2] Caseley, P., Tudor, N., O'Halloran, C., 2003. The case for an evidence based approach to software certification. Safety standards review committee, UK Ministry of Defence.

[3] Dawkings, S., Kelly, T., 1997. Supporting the use of COTS in safety critical applications. In: IEE Colloquium on Cots and Safety Critical Systems. No. Digest No. 1997/013. IET.

[4] de Lemos, R., Saeed, A., Anderson, T., 1998. On the Integration of Requirements Analysis and Safety Analysis for Safety-Critical Systems. Tech. Rep. CS-TR:629, http://citeseer.ist.psu.edu/536230.html, University of Newcastle upon Tyne.

[5] DO-178B, 1992. Software considerations in airborne systems and equipment certification, rtca inc. and eurocae.

[6] DoD, 10 February 2000. MIL-STD-882D Standard Practice for System Safety.

[7] Ellis, A. F., 1995. Achieving Safety in Complex Control Systems. In: Anderson, T., Redmill, F. (Eds.), Achievement and Assurance of Safety, Safety-critical Systems Symposium. Springer-Verlag, Brighton, UK.

[8] Fenton, N., Littlewood, B., Neil, M., Stringini, L., Sutcliffe, A., Writght, D., 1998. Assessing dependability of safety critical systems using diverse evidence. IEE Proceedings - Software 145 (1), 35–39.

[9] Ge, X., Paige, R., McDermid, J., 2010. An iterative approach for development of safety-critical software and safety arguments. In: Proceedings of AGILE Conference 2010. pp. 35–43.

[10] Gerstinger, A., Schedl, G., Winkelbauer, W., 2002. Safety versus Reliability: Different or Equal. In: 20th International System Safety Conference. System Safety Society, Denver, Colorado, USA, pp. 393–400.

[11] Hall, J. G., Rapanotti, L., 2009. Assurance-driven design in Problem Oriented Engineering. International Journal On Advances in Systems and Measurements 2 (1), 119–130.

[12] Hall, J. G., Rapanotti, L., 2012. Software engineering as the design theoretic transformation of software problems. Innovations in Systems and Software Engineering 8 (3), 175–193, DOI 10.1007s11334-011-0171-2.

[13] Hall, J. G., Rapanotti, L., 2016. A design theory for software engineering. Technical Report TR2016/01, Department of Computing and Communications, The Open University, Walton Hall, Milton Keynes, MK7 6AA.

[14] Hall, J. G., Rapanotti, L., Jackson, M., March 2008 2008. Problem Oriented Software Engineering: solving the package router control problem. IEEE Trans. Software Eng. 34 (2), 226–241.

[15] Hall, J. G., Rapanotti, L., Jackson, M. A., 2006. Problem oriented software engineering. Tech. Rep. 2006/10, The Open University.

[16] Hammond, J., Rawlings, R., Hall, A., 2001. Will it work? In: Proceedings of the 5th IEEE International Symposium on Requirements Engineering. IEEE Computer Society Press.

[17] IEC, March 2000. Standard 61508. Functional Safety of Electrical / Electronic / Programmable Electronic Safety-Related Systems.

[18] IEC, 2003. International Standard 61511. Functional Safety / Safety instrumented systems for the process industry sector.

[19] Jackson, M., 2001. Problem Frames: Analyzing and Structuring Software Development Problems. Addison-Wesley Publishing Company.

[20] Kelly, T., 2004. A systematic approach to safety case management. In: Proceedings SAE 2004 World Congress. Detroit, US.

[21] Kelly, T., Weaver, R., 2004. The goal structuring notation–a safety argument notation. In: Proceedings of the Dependable Systems and Networks (DSN) 2004 Workshop on Assurance Cases.

[22] Leveson, N., 1995. Safeware: system safety and computers. Addison-Wesley, Reading, Mass.

[23] Lutz, R. R., 2000. Software engineering for safety: a roadmap. In: ICSE '00: Proceedings of the Conference on The Future of Software Engineering. ACM, New York, NY, USA, pp. 213–226.

[24] Mannering, D., 2009. Problem Oriented Engineering for software safety. Ph.D. thesis, Open University.

[25] Mannering, D., Hall, J. G., Rapanotti, L., September 2007. Safety process improvement with POSE & Alloy. In: Saglietti, F., Oster, N. (Eds.), Computer Safety, Reliability and Security (SAFECOMP 2007). Vol. 4680 of Lecture Notes in Computer Science. Springer-Verlag, Nuremberg, Germany, pp. 252–257.

[26] Mannering, D., Hall, J. G., Rapanotti, L., 2007. Towards normal design for safety-critical systems. In: Dwyer, M. B., Lopes, A. (Eds.), Proceedings of ETAPS Fundamental Approaches to Software Engineering (FASE) '07. Vol. 4422 of Lecture Notes in Computer Science. Springer Verlag Berlin Heidelberg, pp. 398–411.

[27] Mannering, D., Hall, J. G., Rapanotti, L., 5-7 February 2008. Problem oriented safety process improvement. In: Proceedings of the Safety-critical Systems Symposium 2008. Bristol, UK.

[28] Markov, G., Hall, J. G., Rapanotti, L., 19-20 February 2015. An engineering framework for dealing with change problems: theoretical underpinnings and initial evaluation. In: Proceedings of the 15th International Conference on Knowledge, Culture and Change in Organizations and the Organization Knowledge Community. Berkeley, California.

[29] Martino, P. A., Muniak, C., 2002. The Role of System Safety Engineering in Product Safety. In: 20th International System Safety Conference. System Safety Society, Denver, Colorado, USA, pp. 439–447.

[30] McDermid, J., Nicholson, M., Pumfrey, D., Fenelon, P., 1995. Experience with the application of HAZOP to computer-based systems. In: Proceedings of the Tenth Annual Conference on Systems Integrity, Software Safety and Process Security. Computer Assurance (COMPASS'95). IEEE, pp. 37–48.

[31] Menon, C., Kelly, T., 2010. Managing safety requirements across supply chains. In: System Safety 2010, 5th IET International Conference on. IET, pp. 1–6.

[32] Nkwocha, A., Hall, J. G., Rapanotti, L., 2013. Design rationale capture for process improvement in the globalised enterprise: an industrial study. Journal of Software and Systems Modeling 12 (4), 825–845, online first (`http://www.springerlink.com/content/d45x17g438833069/`).

[33] Oates, B. J., 2007. Researching Information Systems and Computing. Sage South Asia Ed.). New Dellhi: Sage South Asia Publication. ISBN.

[34] O'Halloran, M., 2012. An integrated approach to software development for safety-critical systems using Problem Oriented Engineering. Master's thesis, The Open University.

[35] Overton, J., Hall, J. G., Rapanotti, L., May 2010. Middle-out design: A proposed best-practice for GEOSS design. Technical Report 2010/10, The Open University. URL `http://computing-reports.open.ac.uk/2010/TR2010-10.pdf`

[36] Profeta, J. A., Andrianos, N. P., Yu, B., November 1996. Safety-Critical Systems build with COTS. IEEE Computer, 54–60.

[37] Rogers, G. F. C., 1983. The Nature of Engineering: A Philosophy of Technology. Palgrave Macmillan.

[38] SAE, 1996. ARP4761: Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment. Tech. rep.

[39] Schedl, G., Gerstinger, A., Ronach, A., 2001. Safety Management Approach for the Voice Communication System According to Eurocontrol's Air Navigation System Safety Assessment Methodology. In: 19th International System Safety Conference. System Safety Society, Huntsville, Alabama, USA.

[40] Scott, J. A., Preckshot, G. G., Gallagher, J., October 1995. Using Commercial-Off-the-Shelf (COTS) Software in High-Consequence Safety Systems. In: IEEE 1995 Nuclear Science Symposium and Medical Imaging Conference. San Francisco, California.

[41] UK-MoD, 1 June 2007. Safety Management Requirements for Defence Systems Part 1 Requirements. Defence Standard 00-56 Issue 4, MoD.

[42] Weick, K. E., 1995. Sensemaking in Organisations. Thousand Oaks, Calif. Sage Publications.

[43] Yin, R., 2003. Case study research: Design and methods (2nd ed.). Thousand Oaks, CA: Sage Publishing.