JUSTIFYING PLC-BASED APPLICATIONS WITH LIMITED COOPERATION FROM PLATFORM SUPPLIER – THE COGS APPROACH

Gareth Fletcher and Sofia Guerra
Adelard LLP
24 Waterside, 44-48 Wharf Road
London N1 7UX, United Kingdom.
{gtf,aslg}@adelard.com

ABSTRACT

Several control and monitoring applications are implemented using commercial-off-the-shelf (COTS) PLCs that were not necessarily developed according to nuclear standards. The UK nuclear regulatory regime requires that a safety case be developed to justify and communicate their safety. Typically, the assessment of COTS components has been done with a focus on standards compliance – compliance to accepted practice was deemed to imply adequate safety. However, there may be a number of difficulties with justifying COTS products related to limited knowledge of the internal structure of the components or their development processes, especially when the supplier of the PLC platform is not willing to provide the necessary information to complete a compliance case.

This paper describes a claim-based approach to the justification of COTS PLC components using Cogs, developed in a project funded by the UK nuclear industry. The approach

- focuses on the behaviour of the system rather than on the process followed to develop the PLC platform
- structures the justification around behaviour attributes (such as functionality, performance and reliability) and considers them in terms of the application and/or platform
- uses information about the platform that is likely to be publicly available from the supplier

Key Words: Safety justification, commercial-off-the-shelf components, PLCs

1 INTRODUCTION

COTS components are increasingly used in nuclear Instrumentation and Control (I&C) applications. While there are several commercial benefits in the use of COTS components, there are also several challenges and concerns with regards to their safety demonstration and justification.

Traditionally, COTS components have been justified by attempting to show compliance with relevant development standards. This standards-based approach works well in stable environments where best practice is deemed to imply adequate safety and the components were developed according to the relevant standards. However, they are often criticized for being highly prescriptive and impeding the adoption of new and novel methods and techniques. Standards-based approaches to justification are also inadequate where otherwise high-quality systems were developed in accordance with older or different standards, or just meet industrial good practice. This is often the case when industrial components (such as sensors) were not developed specifically for the nuclear industry. In addition, assessing compliance with relevant standards requires detailed information about the development processes followed and the internal structure of the component, which represent important confidential information that the supplier might not be willing to share. Finally, a purely standards-based approach does not necessarily provide direct evidence that the I&C system and its software achieve the behaviour or the properties required to the desired level of reliability.

The safety justification of COTS components is therefore a priority in the UK nuclear industry's research agenda. Research into I&C in the UK is coordinated by the Control and Instrumentation Nuclear Industry Forum (CINIF). The regulator (Office for Nuclear Regulation) and the licensees are all members of the working group. One of the projects funded by CINIF on this topic is entitled "COTS Goal-based Safety assessment" (Cogs), aimed at developing an approach to the safety justification of COTS products. A diverse range of products falls within the scope of Cogs, including:

- Devices dedicated to a single function (e.g., measurement and alarm annunciation instruments).
- User-programmable and control equipment (e.g., PLCs).
- Certain software-only products, such as special-purpose operating systems.

Cogs is a claim-based approach, which uses the Claims-Arguments-Evidence (CAE) framework. The key advantage of a claim-based approach is that there is considerable flexibility in how the claims are demonstrated, since different types of arguments and evidence can be used as appropriate. For example, there may not be much product development evidence available for a component, but there may instead be extensive field experience and records of field-reported faults that demonstrate reliable operation. This might be used as an alternative means of demonstrating adequate product quality and compliance with specified behaviour [1].

This paper follows from a previous NPIC-HMIT paper where we described the Cogs approach [2]. It focuses on applying Cogs to the justification of a PLC-based application where we assume that no cooperation from the PLC platform supplier is available, i.e., only publicly available information can be obtained to justify the platform. We also assume that full cooperation from the application developer is available, and therefore access to design and verification records of the application are available. The approach is to be applicable to PLC-systems implementing a category C function [3].

2 BACKGROUND: CLAIM-BASED JUSTIFICATION APPROACHES

The Claims-Arguments-Evidence (CAE) approach to structuring safety justifications was developed in the EU-sponsored research project SHIP [4]. The Adelard ASCAD manual [5] describes the idea of separating claims, arguments and evidence, and provides a graphical notation to summarize and communicate the justification. The approach has subsequently been refined by application to systems in the defence, nuclear and medical sectors. It is now accepted by the nuclear industry in a number of countries including the UK. The common position document produced by international nuclear regulators on licensing safety critical software [6] also suggests the use of CAE.

There is considerable standardization work on structured cases and CAE and activities internationally in a number of sectors. In particular, ISO/IEC 15026-2 [7] provides a definition of the CAE concept, drawing on Adelard's work.

The key elements of the CAE approach are the following:

- **Claims** are statements of something to be true, with associated conditions and limitations. They are typically statements about a property of the system or some subsystem, or about the development approach used.
- **Evidence** is used as the basis of the justification of the claims. Evidence consists of established facts used as the basis of the justification of the claims. Sources of evidence may include the design, the development process, prior field experience, testing or source code analysis.

- **Arguments** link the evidence to the claim, or link claims to other, more specific, claims. They are the "statements indicating the general ways of arguing being applied in a particular case and implicitly relied on and whose trustworthiness is well established" [8], together with the validation for the scientific and engineering laws used.

The idea is that claims can be broken down into smaller, more readily justified, sub-claims. This process is called *decomposition*. There are a number of types of decomposition, such as:

- **Architectural decomposition**, where a claim about the system is decomposed into sub-claims about its components and their interconnections.
- **Functional decomposition**, where a system-level function is partitioned into sub-functions.
- **Attribute decomposition**, where a claim about the behaviour of the system is decomposed into sub-claims about different aspects of the behaviour.

To help visualize the whole claim tree and the interaction between its parts, a graphical notation can be used showing shapes representing claims, arguments and evidence connected with arrows to indicate where evidence is used to support arguments, and where arguments are used to support claims -Figure 1.

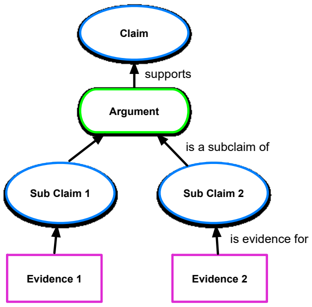

Figure 1: Example of a typical CAE structure in a safety case

3 THE COGS APPROACH

The Cogs approach [2] was developed for the specific context of nuclear applications where a COTS product is a component within a wider I&C system. Cogs is structured around gathering evidence of the intended behaviour of the product and comparing it with available evidence supporting its actual behaviour.

Cogs is generally applicable to several types of product. It aims to support the top-level claim that the component behaves as intended on the basis that certain conditions are met (e.g., the ambient temperature and power supply conditions of the component are met). The main aim is to show that

- the described behaviour and functionality of the product is sufficient to understand its intended behaviour and required properties
- the product behaves according to this description throughout its intended lifetime

The Cogs approach consists of a set of top-level claims to be justified and guidance on how to expand and justify these top-level claims, as illustrated in Figure 2. The four top-level claims are the following:

- Claim 1: the behaviour and functionality of the component are documented adequately
- Claim 2: the component behaves according to its documentation when deployed
- Claim 3: the component will carry on behaving according to its documentation

- Claim 4: sound development process and design principles were followed

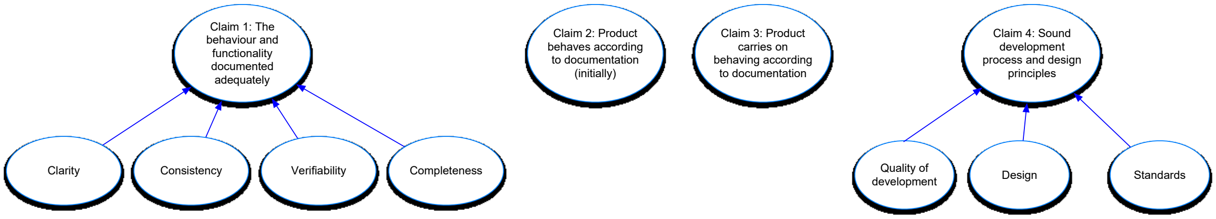
Claim 1: The behaviour and functionality documented adequately
Clarity
Consistency
Verifiability
Completeness
Claim 2: Product behaves according to documentation (initially)
Claim 3: Product carries on behaving according to documentation
Claim 4: Sound development process and design principles
Quality of development
Design
Standards

Figure 2: Cogs claims and top-level decomposition

4 USING COGS TO JUSTIFY PLCS

This section describes an approach to justify a PLC-based system, i.e., a system based on a PLC platform with a specific application loaded. The approach is based on a high-level view of the PLC's components and the behavioural attributes are defined on the PLC as a whole (i.e., the PLC with the running application). We assume that the PLC-based system performs a category C function and that it is implemented within a single PLC CPU.

To justify that the PLC-based system performs as required by the application, we need to establish the four top-level Cogs claims listed in Section 3. The sections below expand on how to justify each of the top-level claims for a PLC-based system.

4.1 Claim 1: The behavior of the system is described adequately

The description of the behaviour is adequate when it is

- **complete** – does not leave out any aspects of the behaviour that are relevant to the application
- **clear** – easily understood and not ambiguous
- **verifiable** – makes assertions that are possible to confirm via testing or analysis
- **consistent** – not contradictory

In general, to provide sufficient detail to construct the justification, we need descriptions of

- **the full behaviour of the component** – including the functions provided, performance attributes (e.g., time response) and dependability attributes (e.g., reliability and failure recovery)
- **the needs of the component when it is deployed** – for example, physical environment constraints, such as temperature, vibration and humidity limits, or interfacing requirements such as connection types, protocols and power

The first item on behaviour can be developed by an attribute decomposition (see Section 2), breaking behaviour into a number of relevant behavioural properties. These include sub-claims for

- **functionality** – the functional behaviour, including all the functions and operating modes
- **performance** – characteristics defining the ability to achieve the intended functions, such as accuracy and time response
- **dependability** – including security, failure integrity, failure recovery, operability, robustness, and reliability

The Claim 1 decomposition is illustrated in Figure 3.

The needs of the component when it is deployed – the conditions under which the described behaviour is guaranteed – may include **required resources** (e.g., support libraries, processor power, memory space, communications bandwidth), **environmental constraints** (e.g., temperature, EMI, humidity) or **operational and maintenance assumptions** (e.g., periodic calibration, correct probe connection).

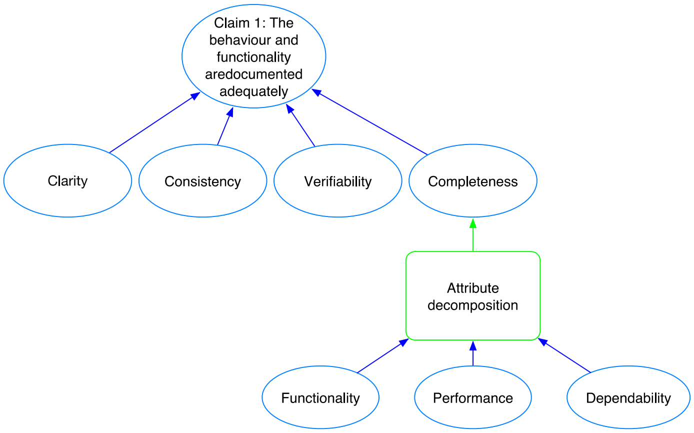

Figure 3: Claim 1 - The description of the behaviour is adequate

4.2 Claim 2: The system behaves according to its documentation (initially)

Once there is an adequate description of the behaviour, the PLC-based system will be assessed to see whether the behaviour actually implements this description.

In Claim 1, the claim of description completeness is structured according to the behavioural properties that describe the behaviour of the PLC-based system. Accordingly, an argument for the system behaving in agreement with this description can be decomposed in the same way. For each property, a claim is made that the behaviour achieved by the device is the same as that claimed in the description.

However, we are assuming that the evidence to support the platform and behaviour of the engineering tools will be based on publicly available information or be generated by the assessor, while evidence to support the application is readily available. Therefore, a different approach needs to be taken for the platform from that of the application.

Taking into account the (lack of) availability of information, the suggested strategy for the justification of the behaviour of the PLC-based system is as follows:

1. Review the attributes of Claim 1 and define
    a. which attributes can be shown at application logic level on its own (e.g., functionality) assuming generic behaviour of the platform;
    b. which attributes need to be shown by considering both the application logic and the behaviour of the platform (e.g., timing);
    c. Which attributes need to be shown at platform level on its own (e.g., possibly accuracy, behaviour following a power supply interruption).
2. Show that the application logic behaves as required by considering each of the attributes or requirements that are shown at the application level (as defined in step 1 above). This can be done by a combination of techniques, including testing, simulation, modelling or analysis. (Claim 2.1)

3. Show that the platform satisfies the requirements identified in step 1.b and 1.c. (Claim 2.2.1)
4. Argue appropriate reliability and correct execution of the application by the platform. (Claim 2.2.2)
5. Review the role and impact of the engineering tools in the execution of the application and justify it. (Claim 2.2.3)

Claim 2 is illustrated in Figure 4.

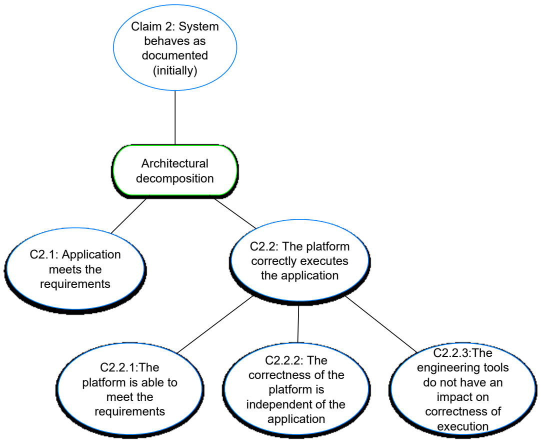
Claim 2: System behaves as documented (initially)
Architectural decomposition
C2.1: Application meets the requirements
C2.2: The platform correctly executes the application
C2.2.1:The platform is able to meet the requirements
C2.2.2: The correctness of the platform is independent of the application
C2.2.3:The engineering tools do not have an impact on correctness of execution

Figure 4: Claim 2 - The description of the behaviour is adequate

4.3 Claim 3: System behaves according to its documentation (over the lifetime)

We establish in Claim 2 that the system behaves correctly under ideal conditions at the time of commissioning. However, after installation, any system is subject to changes, both internally and in its operating context (its environment, the operators and their management, the requirements placed on it, etc.). It is not enough to show that the behaviour is as specified; it is also needed to show that the behaviour will continue to be as described over its entire lifetime, in spite of the evolving environment, capabilities and any changes, be them deliberate, planned, accidental or out of user's control.

The consistent behaviour over the component's lifetime depends on

- the correct environment of the component and the continuing fulfilment of the device needs
- the modifiability of the component and how likely faults are to be introduced when the component is modified, either deliberate changes such as calibration, or unintended changes such as damage or vandalism
- changes due to age being rendered benign by corrective or preventative maintenance

4.4 Claim 4: the development process meets key principles

Whereas the previous claims focus on the device and its behaviour, Claim 4 considers other principles that are not directly related to the device behaviour but nevertheless have an important role to play in the safety justification. The principles considered in this claim are related to

- the development process and supporting processes, such as quality assurance processes and configuration management
- design principles
- compliance with identified relevant standards

Given the assumptions on availability of information in this example, the focus of this claim will be on the development process of the application.

5 APPLYING THE APPROACH TO A CASE STUDY

We are currently developing the details of the approach by applying it to a case study within the nuclear industry. We have been working with a UK licensee, who has provided us with the specification of an application that is using a PLC. We have also bought the safety PLC that is planned to be used in this application, to obtain the evidence missing from the publically available documentation was that was necessary to complete the justification.

5.1 Description of the Case Study

In the case study, the PLC is part of a more complex system that controls the environment of a facility that performs both control and protection functions. The system actuates valves and pumps, and perform protection functions by managing interlocks and acting in the presence of faults. The PLC-based system we are using in the case study is only one channel of several that perform different functions in the overall facility.

5.2 Behavioural Attributes

As described in Section 0, we reviewed each of the behavioural attributes and classified them according to whether they would need to be justified at platform level (Plat), application level (App) or both. Table 1 shows the attributes, their classification and some additional comments explaining the attribute.

Attribute	Plat	App	Comments
Functionality	No	Yes	Functionality of the application program logic.
Performance: Accuracy	Yes	No	Accuracy requirements will need to be supported by the accuracy of the PLC components.
Performance: Timing	Yes	Yes	The justification of timing requires consideration of both the application and the platform (and associated engineering tools).
Failure integrity	Yes	Yes	Revealing internal faults is a feature of the PLC as a whole (e.g., internal alarm output), supported by self-monitoring functionality.
Failure recovery	Yes	Yes	Recovering the device from a failure state that occurs from the application or platform.
Security	Yes	No	Security is a system wide attribute, where the PLC as a whole may enforce confidentiality and integrity restrictions on the configuration. For example, prevent readout or modification by unauthorised users, which translate into particular aspects of functionality for the OS and language executive. The application software could contribute to security if features are supported in the development framework.
Operability	No	Yes	Usability is mostly an application issue, but could include any front-panel interface on the PLC itself (labelling of connections or display of detected errors).

Attribute	Plat	App	Comments
Robustness	Yes	Yes	The behaviour outside of normal operating conditions is understood. Mainly related to the platform (power supply, inputs), but also related to how the application logic checks inputs.
Reliability	Yes	Yes	Absence of faults in the application and platform. Correct implementation of function blocks could be linked to IEC 61131-3 compliance, if claimed. Fault detection and tolerance: platform, but could also have recommendations on how to build the application logic. Safe state: platform with respect to the application.
Environmental and operational pre-requests: Non-interference	Yes	No	Non-interference for the hardware refers to electromagnetic compatibility, etc. There may be elements of correct bus interactions for the OS, but for the remaining software, non-interference is not meaningful without an application.

Table 1: PLC Cogs attributes

5.3 Supporting Evidence

As part of the case study, we reviewed the public information available on the PLC platform and programming tools. We were able to use some of the environmental tolerance and accuracy data in the specification sheet to provide evidence for claims on various Cogs behavioural attributes. We were also able to use information about the software programming tools, such as the programming languages being IEC 61131-3 compliant to support the sub-claim on “the engineering tools do not have an impact on the correctness of the execution”. Any SIL certifications that the PLC has accredited have been used within Claim 4 to support the compliance with the standards sub-claim.

5.4 Testing the PLC

Tests can be performed on the PLC platform to help provide evidence to justify claims where public information is not available, such as

- some of the behavioural attributes, including performance, security, failure integrity and failure recovery (Claim 2)
- the correctness of the platform tools - programming software, compiler, program loader (Claim 2)
- design prevents accidental or unauthorised changes (Claim 3 and the security behavioural attribute)
- Platform’s behaviour after a power failure (Claim 2 and 3)

As part of the project, we have purchased a safety PLC that is used within the UK nuclear industry. From our installation and initial testing studies, we were able to identify several useful features built into the PLC platform:

- in safety mode the I/O modules can enter a predefined safe state upon encountering an error (fail-safe)
- the hardware configuration is stored within the PLC, which is checked at power up; the PLC will not go into run mode unless the stored configuration matches that in the PLC hardware configuration

- module configuration parameters can be password protected (configuration integrity)
- the logic of the program can be viewed live when the PLC is in run mode with the programming device connected (application testing)
- there is a checksum of the application blocks and hardware configuration; however we are not sure of the exact contents (configuration integrity)
- messages are logged in the PLC message buffer, such as power failure or restarts (diagnostics)

5.4.1 Example: 1 out of 2 voting evaluation test

1 out of 2 (1oo2) evaluation is often used in safety applications to provide redundant digital I/O channels. This protects the system if one input channel fails, such as in the case of a wire break or relay failure. It also allows diagnostics such as time discrepancy testing between the channels to be performed to diagnose input channel faults.

Using the safety PLC that we had purchased, we set up a simple set and reset switch using two input channel pairs, with the channel pair discrepancy time set to 10 ms. The ladder logic diagram of this simple test is shown in Figure 4.

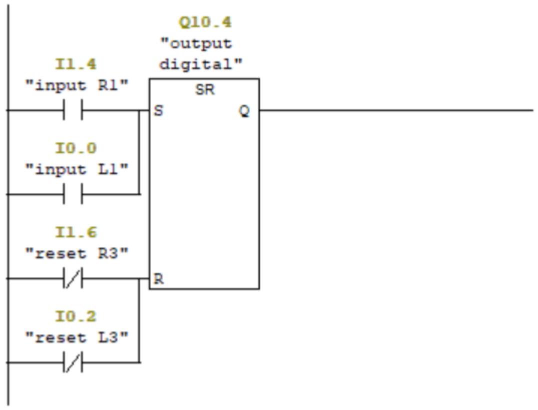

Figure 4: 1oo2 evaluation test

First, we tested that the output relay activated when both I0.0 and I1.4 were energized together and deactivated when I1.6 and I.02 deactivated. This worked with no diagnostic fault being recorded by the PLC. Then we tested the same procedure with the digital input I1.4 disconnected (simulating a wire break). I0.0 was energized but a diagnostic message was recorded and the input module displayed a visual fault light warning while I0.0 was energized, representing the detection of a time discrepancy fault. Since the default state for the reset digital inputs is energized, we only had to disconnect one of the input wires to I1.6 to test the time discrepancy fault on the reset input. Overall, this showed that the PLC could detect both synchronous and asynchronous digital channel discrepancy faults. It also alerted the user both visually on the module and in the PLC diagnostic message stream, providing evidence for the fault detection and fail-safety behavioural attributes.

6 CONCLUSIONS

This paper describes the Cogs approach to justifying the use of COTS components in nuclear safety applications and its use to the justification of PLC-based systems. Cogs is based on a flexible, claim-based framework and aims to contribute a number of benefits:

- Flexibility when compliance with standards cannot be demonstrated.

- Focus on behaviour rather than on process.
- Addresses lifetime issues and integrates well with the overall I&C safety case.

In this paper, we described how to use it for justifying a PLC-based application where no cooperation from the platform supplier is established, although we assume that full cooperation with the application developer has been established. The approach

- focuses on the behaviour of the system rather than on the development processes
- structures the justification around behaviour attributes (such as functionality, performance and reliability) and considers them in terms of the application and/or platform
- uses information about the platform that is likely to be publically available

We have been performing a case study to develop the approach. In order to explore what evidence we could obtain if no cooperation from the platform was established, we bought a safety PLC and have been studying what evidence we can obtain from the documentation that is not publicly available but is delivered with the PLC and from testing the PLC platform itself. Although this is still work in progress, we have already identified areas in the case that are possible to justify by testing a PLC platform rather than by usual compliance approach. We have still to explore whether this together with the application specific evidence would be enough to complete a case to a satisfactory degree.

7 ACKNOWLEDGMENTS

This work was funded by the UK Control and Instrumentation Nuclear Industry Forum (CINIF).

8 REFERENCES

1. S. Guerra, P. Bishop, R. Bloomfield and D. Sheridan, "Assessment and Qualification of Smart Sensors," *Seventh International Topical Meeting on Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies (NPIC&HMIT)* (2010).
2. S. Guerra, N Chozos and D. Sheridan, "Justifying Digital COTS Components when Compliance Cannot be Demonstrated – The Cogs Approach", *Ninth International Topical Meeting on Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies (NPIC&HMIT)* (2015).
3. "Nuclear power plants – Instrumentation and control important to safety – Classification of instrumentation and control functions" IEC 62671:2010.
4. P. G. Bishop and R. E. Bloomfield, "The SHIP Safety Case – A Combination of System and Software Methods", *SRSS95, Proceedings of 14th IFAC Conference on Safety and Reliability of Software-based Systems*, Brugge, Belgium, 12–15 September, (1995).
5. "Adelard Safety Case Development Manual", Adelard, ISBN 0 9533771 0 5 (1998).
6. "Licensing of safety critical software for nuclear reactors – Common position of international nuclear regulators and authorised technical support organizations", Bel V, BfS, CNSC, Consejo de Seguridad Nuclear, ISTec, KAERI, KINS, NSC, ONR, SSM & STUK, http://www.onr.org.uk/software.pdf (2018).
7. "ISO/IEC 15026-2:2011, Systems and software engineering – Systems and software assurance, Part 2: Assurance case." (2011).
8. S. E. Toulmin, "The uses of argument", Cambridge University Press (1958).
9. R. Stockham, "Emphasis on safety", E&T Magazine, Issue 02 (2009).