ADELARD

CAE MINI-GUIDE 7: REVIEW AND CHALLENGE

Summary

This document is part of the Declare CAE guidance document set. It contains guidance on the Review and Challenge of CAE justifications.

24 Waterside
44–48 Wharf Road
London
N1 7UX

T +44 20 7832 5850
F +44 20 7832 5870
E office@adelard.com
W www.adelard.com

Authors
Robin Bloomfield
Nick Chozos

Distribution
As per Cinif list in
Adelard D1284v10

Contents

Figures

Tables

1 Introduction

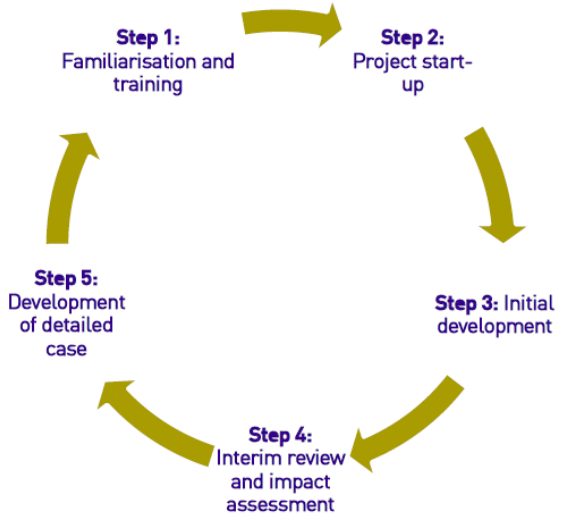
This document is a mini-guide that forms part of the Declare CAE guidance document set and provides guidance on Review and Challenge of CAE justifications.

2 This mini-guide and the CAE document set

The CAE guidance can be seen as having two main components:

1. **CAE process:** The first component describes an overall process made up of five steps (the "CAE process"), explaining the evolution of a justification within an organisation and the activities involved. The process is adaptable and flexible and it may apply differently to different projects depending on the scenarios of use.
2. **CAE mini-guides:** The second part provides specific technical guidance on the underlying concepts, their definition and their application. We have compartmentalised the technical guidance into "mini-guides": small, dedicated sets of guidance each focusing on a particular issue. Each mini-guide contains a concise summary with a short list of the key points and risks and challenges that need to be considered, which is then supported by more detailed guidance.

The CAE process, and the supporting mini-guides, are summarised in Figure 1 below. This document is highlighted (mini-guide 7) in this figure.



	Mini-guide
1.	One-page guidance
2.	CAE concepts
3.	Formulation of top-level claim
4.	Connection rules
5.	The CAE building blocks
6.	Summarising and communication
7.	**Review and challenge**

Figure 1: Summary of the CAE process and supporting mini-guides

The overall CAE process is described in the main CAE guidance document [1]. The process is flexible and adaptable, and depending on the project, only specific phases may be required. The main guide explains the various scenarios of use and how the guidance may apply in different cases. The document also discusses how the mini-guides may be used in different scenarios and at different phases of a project.

Table 1 below illustrates how this mini-guide (Review and Challenge) applies throughout the CAE process.

CAE steps	Review and Challenge
Step 1: Familiarisation and training	Review and preparation

CAE steps	Review and Challenge
Step 2: Project start-up	Identify reviewers and plan review milestones
Step 3: Overview and initial development	Group discussion Initial engagement of stakeholders to provide project context
Step 4: Interim review and safety case impact assessment	Preliminary review by colleagues and stakeholders
Step 5: Development of a more detailed case	Full review by stakeholders, possibly including independent assessor / advisor

Table 1: Relationship of this mini-guide to the CAE process

3 Mini-guide 7: Review and challenge

The process for review and challenge has its own specific guidance but should be applied throughout the CAE process, focussing on different aspects as the safety case evolves. This activity is fundamental to developing a case that we have justified confidence in. The purpose and focus of review can be varied as it can be part of

- development and architecting of the case
- formal confidence building (e.g. independent review)
- formal decision making (e.g. go/no go operational decision)

The focus of the review could either be editorial or be from a range of different technical viewpoints.

3.1 Initial review

An initial review of the case should establish

- an understanding of what it is for and what it concerns
- the range of stakeholders involved
- the systems it concerns
- the status of project in terms of its criticality and the decision making it supports
- the provenance of the case, how the case was produced (e.g. template, brainstorming, copy and paste!)

The next step is to gain an overall impression of the following:

- understand the main claims
- review the architecture of the case
- establish what the key evidence is
- assess the topology, whether it is in normal form, if there are any nodes with an excessive number of subclaims
- appraise the status of completion (e.g. evidence identified but lacking)
- the presentation of the case

3.2 Technical reviews

In reviewing the CAE structure, we should address the V&V of the CAE as well as consider its fitness for purpose and explore other design and assurance options. Questions to be addressed are defined in Table 2 and Table 3.

Question	Discussion
Are the CAE concepts properly applied and is the CAE properly formulated?	Are the claims, arguments and evidence actually claims, arguments and evidence?
Is the structure of the case sufficiently complete?	Does the evidence provide a link to the top-level claim? Are all leaves of the CAE tree either subclaims that are recognised as assumptions or evidence?
Does the case follow the connection rules?	See connection rules mini-guide ([5]) Are any deviations justified?
Are the CAE blocks applied correctly?	Do the CAE blocks comply with their definitions? (see guidance in [6]). For instance, justify that dependability = {reliability, availability} or justify that if system model is A and B then response time of system is response time of A + response time of B
Does the case reflect the real world?	Perform validation – does the case actually reflect the real world? Have the CAE blocks side claims been sufficiently justified? E.g. "Is the system made of A and B (only A and B?)" or "Does the property distribute?"

Table 2: V&V of CAE structure

Question	Discussion
Is the case feasible?	Are the claims realistic? Is it possible to get the evidence at appropriate cost? Are the assumptions well-articulated? Are they valid?
Are there better ways of achieving the same assurance?	Could the case use different types of evidence? Has maximum use been made of evidence? (e.g. operating experience)
Is the impact on other systems and processes acceptable?	Will the case have an unacceptable impact on system design or operation? (e.g. complex architecture from diversity requirements, complex operator actions from not using computer based systems)

Table 3: Optioneering

In the review we should consider

- Using an explicit model based approach to reasoning about the system behaviour.

- Knowledge of known vulnerabilities in systems and previous issues in assurance justifications. It is important to broaden the appreciation of what can go wrong. This could be captured in a variety of checklists.
- Applying checklists and prompts specific to the CAE Blocks.
- Applying hazard analysis techniques to the case itself to assess impact of issues with it and focus the review.
- Developing a diverse case to explore the validity of the claim. For example, an "anti-case" could be developed on why the system does not have the claimed property.
- Whether the case reflects the assurance principles (adapted from IAEA) in Table 4.

• Effective understanding of the hazards and their control should be demonstrated. • Intended and unintended behaviour of the technology should be understood. • Multiple and complex interactions between the technical and human systems to create adverse consequences should be recognised. • Active challenge should be part of decision making throughout the organisation. • Lessons learned from internal and external sources should be incorporated. • Justification should be logical, coherent, traceable, accessible and repeatable with a rigour commensurate with the degree of trust required of the system.

Table 4: Assurance principles

3.3 Implicit and explicit models

The CAE structure will address a property of a system or organisation and as such relies on explicit or implicit models to give it meaning e.g. our ideas of what a "crane' is.

A model represents the system in a way that is relevant to the claim or property being justified[1]. If the model justifies the claim being assessed, we can focus on the residual doubt surrounding the evidence we have used and the validity of the model.

As part of deciding whether we have done enough we might address whether the model has been applied correctly, and whether it is valid. These "stopping rules" then concern the following questions:

- Does the model capture the required behaviour?
- Is it based on sound principles?
- Has it been applied correctly?
- Does it provide adequate results?
- Is it valid?
- Does it capture all credible fault types?
- Does it contradict other evidence (and vice versa)?

So the flow of the claim process is as follows:

1. Develop an appropriate model.
2. Increase the detail and rigour of the model until a firm judgement can be made about the claim.
3. Check that the model has been applied correctly.
4. Check the trustworthiness of the evidence used.

[1] We would have different models for showing we can serve drinks in first class by the time a plane lands compared to one for assessing its aerodynamic stability.

We illustrate the stopping rule by exploring how we might develop a case for the time response of a smart sensor. At the system level, suppose we have a temperature that has to be measured and transmitted to a controller. As mentioned above, claim decomposition can be driven by a number of partitioning approaches by architecture, attributes or activities. In this case we might "concretise" the attribute *timeliness* as a response time for these abstract signals, and consider a separate claim for accuracy. The response time for the signal would then be apportioned to different components – an architectural decomposition – and we would arrive at a specification for the device in terms of its concrete inputs and outputs. So

- the abstract attribute *timeliness* would be used to prompt the definition of temperature response time
- the system response time would be refined to produce a smart device response time
- the system temperature would be related to the measured signal

We would then apportion the time response to different parts of the smart device (the A/D conversion, the output D/A and the main processing) and arrive at a software response time requirement.

In order to justify the response time, we would need a model of resource usage for the software. Our first attempt might be a simple yet conservative model that we could use to try to show that the response time is deterministic by design and is within the bounds. We could then analyse the inadequacies with this model and develop a more detailed justification.

We found in this case study that a focus on justifying the claim from the device requirements and design might miss possible failure modes and sources of timing problems. For example, in examining a real device we might find that part of the lookup table code uses loops with different numbers of iterations in a binary search – not strictly deterministic but expected to be upper bounded (so accuracy and timing become related, because a bigger lookup table will provide more accurate results). Demonstrating that this is satisfactory from the design point of view requires access to the code or a very detailed pseudo-code like description of the algorithm used and therefore raises the related issue of how much can be done "black-box".

Table 5 shows how the stopping rule questions have been interpreted in this example.

Question	Comment
Does the model capture the required behaviour?	The model is about timing and not some other aspect. It provides an upper bound on the execution time.
Is it based on sound principles?	Can we find examples in the literature? Yes for "Worst Case Execution Time" in general but not for this specific simplified application of it.
Has it been applied correctly?	The results have been reviewed and checked with independent diverse calculations.
Does it provide adequate results?	Is the bound calculated within the required response time? If not, understand why. Remove some approximations, detail model or abandon approach and accept negative result. Does the model increase our understanding and insights?
Is it valid? Does it capture all credible fault types?	Are the assumptions credible? Look for any credible mechanisms that would lead to significant time delays and stop when we can show that these are not present or have a quantified impact.
Does it contradict other evidence (and vice versa)?	Is the model consistent with test results? Can we explain the degree of pessimism? Does it explain the fastest response time?

Table 5: Example questions for smart sensor devices

3.4 Presentation of the case

The review should also consider how the case is presented and whether the CAE should be grown, pruned or condensed.

Reductionism not appropriate

If you are working top down to develop the CAE, there may be many forms of divide and conquer by applying the CAE decomposition block to grow and detail the structure. At some level of detail this tactic will cease being useful as reductionism is not appropriate (e.g. is it a system level or emergent property that cannot be assessed in terms of components).

Graphical is not helpful

There are some aspects where the graphical approach might not be appropriate and that linking or including tables is a better approach. For example, representing a hazard log graphically may provide an unhelpfully large cauliflower-like figure. Is there a good balance between graphical and narrative?

Pruning the CAE structure

The CAE structure might be pruned for reasons of relevance and appropriate presentation.

- As the case develops, claims and information that were thought to be relevant may not be needed. Is there unnecessary clutter in the CAE?
- What are the essential aspects to the case? Would some aspects be better presented in a narrative or as a set of assumptions?

Addressing the needs of different stakeholders

There may be a need for different viewpoints on the CAE by different stakeholders and different levels of detail. Have these needs been addressed? A companion mini-guide addresses summarising and communication [7].

4 Summary guidance

- The process for review and challenge has its own specific guidance as presented in this document, but should be applied throughout the CAE process, focussing on different aspects as the CAE structure evolves.
- We may review a CAE justification taking a variety of perspectives (e.g. correct use of CAE, completeness of arguments, if it's fit for purpose etc)
- Guidance in specific mini-guides can be used to review each of the corresponding subject areas
- This mini-guide contains a number of questions that can be used as checklists when performing a CAE review

5 Acknowledgements

We would like to thank Sellafield Ltd and ONR for their high level of engagement with the project, and particularly Sellafield Ltd for their support and involvement in the project workshops.

This deliverable draws on a number of sources developed in earlier Cinif, SSM and Adelard projects (and in particular.

6 Bibliography

6.1 CAE main-guide

[1] Bloomfield R, Chozos N, Declare: CAE main guidance and process, Adelard document reference D/1284/43195/2, version v1.0, April 2020

6.2 CAE mini-guides

[2] Bloomfield R, Chozos N, CAE mini-guide 1 – One-page guidance, , Adelard document reference W/3104/43195/1, version v1.0, April 2020

[3] Bloomfield R, Chozos N, CAE mini-guide 2 - CAE concepts, Adelard document reference 2, Adelard document reference W/3104/43195/1, version v1.0, April 2020

[4] Bloomfield R, Chozos N, CAE mini-guide 3 – Formulation of top-level claim, Adelard document reference 2, Adelard document reference W/3104/43195/1, version v1.0, April 2020

[5] Bloomfield R, Chozos N, CAE mini-guide 4 - Connection rules, Adelard document reference W/3104/43195/1, version v1.0, April 2020

[6] Bloomfield R, Chozos N, CAE mini-guide 5 - The CAE building blocks, Adelard document reference W/3104/43195/1, version v1.0, April 2020

[7] Bloomfield R, Chozos N, CAE mini-guide 6 – Summarising and communication, Adelard document reference W/3104/43195/1, version v1.0, April 2020

[8] Bloomfield R, Chozos N, CAE mini-guide 7 - Review and challenge, Adelard document reference W/3104/43195/1, version v1.0, April 2020 (this document)