ADELARD
part of nccgroup

CAE MINI-GUIDE 8: DEFEATERS

Summary

This document is part of the Declare CAE guidance document set. It contains guidance on Defeaters.

65 Curzon Street
London
W1J 8PE

T +44 161 209 5200
E office@adelard.com
W www.adelard.com

Authors
Robin Bloomfield
Elitza Karadotcheva
Ben Phillips
Panayiota Sketsiou

Copyright © 2024
NCC Group plc.
All rights reserved.

Contents

1 Introduction 3

2 This mini-guide and the CAE document set 3

3 Mini-guide 8: Defeaters 4
- 3.1 Background and definitions 4
  - 3.1.1 Types of defeaters and stages of defeat 5
  - 3.1.2 Impact on the CAE case 5
- 3.2 Defeaters identification 6
  - 3.2.1 Brainstorming and "hazard analysis" 6
  - 3.2.2 Use of guidance and standards 10
  - 3.2.3 Fallacies 11
- 3.3 Applying and addressing defeaters 14
  - 3.3.1 Disjunctive defeater blocks 14
  - 3.3.2 Addressing defeaters 16

4 Acknowledgements 17

5 Bibliography 18
- 5.1 CAE main-guide 18
- 5.2 CAE mini-guides 18

Figures

Figure 1: Summary of the CAE process and supporting mini-guides 3
Figure 2: Example brainstorm canvas 8
Figure 3: Example prompt tree from IAEA Guidance [3] 11
Figure 4: An example case showing conjunctive decomposition, and disjunctive decomposition with defeaters (adapted from [4]) 15

Figure 5: Exact/eliminative defeater example 16
Figure 6: Addressed defeater 17

Tables

Table 1: Relationship of this mini-guide to the CAE process 4
Table 2: Example outline of a defeater workshop 7
Table 3: Brainstorming issues 9
Table 4: Hazard identification in Argument steps in a CAE case 10
Table 5: Common fallacies and the associated symptoms 14

CONFIDENTIALITY, INTELLECTUAL PROPERTY RIGHTS, AND DISCLAIMER STATEMENT
The information contained in this Report has been produced on behalf of EDF Energy Nuclear Generation Limited, Nuclear Decommissioning Authority (Sellafield Ltd.), AWE plc, Urenco UK Ltd., Devonport Royal Dockyard Limited, Rolls-Royce Submarines Ltd. and Rolls-Royce SMR ("the Parties"). This Report is the property of EDF Energy Nuclear Generation Limited ("the Lead Party") who hereby grants each of the other Parties and their successor companies, an irrevocable royalty-free, non-exclusive licence to EDF Energy Nuclear Generation Limited's rights to the Intellectual Property generated in the Report. This is in accordance with Clause 10 of the CINIF Research Agreement. This information is to be held strictly in confidence within the Parties and duly authorised recipient organisations including the Office for Nuclear Regulation, the Health and Safety Executive, Government Departments, or non-Parties with a support contract to assess a Party's safety case. No disclosure is to be made to any other third party without the written agreement of the Lead Party and is to be used solely for the purposes sanctioned by the Parties.

DISCLAIMER
The views expressed in this Report are those of the author(s) and do not necessarily represent the views of the members of the Parties. The Parties do not accept liability for any damage or loss incurred as a result of the information contained in this Report.

1 Introduction

This document contains guidance on defeaters. A defeater potentially refutes or undermines our confidence in a case: it attacks the justification being made by identifying possible problems with the claims, arguments or evidence that make up the case. Defeaters of a case are analogous to hazards of a system.

The explicit use of defeaters in cases is relatively new in industry, so the guidance is less mature compared to other Declare guidance. Therefore, we particularly appreciate any feedback.

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 approach is flexible, adaptable, and will apply differently to different 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 8). A list of the full guidance document set can be found in Section 5.2.

	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
8.	**Defeaters**

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

The overall CAE process is described in the main CAE guidance document [11]. 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 (CAE concepts) applies throughout the CAE process.

CAE steps	Defeaters
Step 1: Familiarisation and training	Review and preparation.
Step 2: Project start-up	Sources of defeat can be identified at this stage including applicable standards and hazards and experience with previous justifications. Defeater probing questions defined.
Step 3: Overview and initial development	A strawman case could be developed and high-level, non-formalised defeaters identified while optioneering. Initial potential defeater list defined. Consider initial defeater workshop. Case evolves as defeaters are addressed.
Step 4: Interim review and safety case impact assessment	Defeater identification and application of defeater nodes to the structure. Potential impact of defeaters assessed, and overall mitigation strategy defined. Further defeater workshop with colleagues and stakeholders.
Step 5: Development of a more detailed case	Identified defeaters are successfully addressed or top-level claim unsupported and case fails. Review of update of case based on earlier defeater workshops.

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

3 Mini-guide 8: Defeaters

In this section we provide guidance on defeaters: what they are, how they can be identified, and how they can be applied in practice and addressed.

3.1 Background and definitions

The formal concept of defeaters originates in epistemology and argumentation theory, where the focus in particular is the justification of a certain belief and the defeasibility of that belief [1][7]. A defeater potentially refutes or undermines our confidences in that belief.

Doubts in safety cases, particularly in complex and challenging contexts such as developing a CAE justification for a C&I system, can be reasonably expected to exist. They can be due to emergent changes to the system, lack of justification for the C&I development approaches used, gaps in our scientific knowledge, simple omissions, or even bias. The application of defeaters can increase our confidence in the safety case by either demonstrating that weaknesses are not present, or by improving the CAE or the system itself, by addressing weaknesses identified. The application of defeaters can become complex with the need to consider defeating defeaters in a chain of defeat, self-defeating arguments, and collective defeat. There is wide philosophical and technical literature on defeaters and the related topics of argumentation dialogues.

For those with a safety background, it may be helpful to see defeaters of a case as analogous to hazards of a system. Just as with hazards, we ask what could go wrong with the case and how we might remove these

by design of the case itself or by mitigating the "hazards". As with hazard analysis, we also need to learn from experience about the range and types of defeaters.

The motivation for deploying the defeater concept systematically is that it can

1. provide a mechanism for sharing and building on past experience of cases
2. allow the user to build confidence in the case
3. identify potential issues with the case early in the project
4. support the judgment that enough has been done on a case
5. justify why the top-level claim is not supported

3.1.1 Types of defeaters and stages of defeat

There are many types of defeaters proposed in the literature, but for safety cases we consider two main types based on Pollock [1]:

- Rebutting: Reasons for believing a directly opposite conclusion, i.e., "claim is false".
- Undercutting: An undermining position, reasons for doubting that the evidence or inference does indeed support the claim, e.g., "argument does not apply in this situation".

In practice we might not know at initial identification which of these classes the defeater is in.

In the final assurance case, all known defeaters should be addressed and no new defeaters can be imagined: the case is then said to be indefeasible [2][4]. To achieve this, a systematic search for defeaters needs to be applied and documented throughout the development of the assurance case. The identified defeaters need to be defeated themselves through further analysis.

It is helpful to consider three stages of defeat:

1. initial doubts – where we have identified an area of concern, e.g., "this argument step might rely on independence assumptions, and we know these are hard to justify"
2. exploratory defeater – when the specific issue has been identified, e.g., "the claimed independence is not justified, this undermines the reliability claim"
3. addressed or accepted defeat – when a defeater has been dealt with e.g. "lack of independence has been taken into account by reducing the overall reliability claim and increasing diversity"

3.1.2 Impact on the CAE case

An indefeasible case requires all known defeaters to be resolved. Therefore, the impact of each defeater on the overall argument of the case needs to be understood and assessed individually.

In the nuclear industry, CAE is usually used to present a positive case that, for example, we are confident that the C&I system meets its relevant safety requirements. Defeaters then attack this positive case and can impact the case structure in two ways:

- New claims, arguments, evidence, and narration is added to make sure that the issues raised by the defeater, if relevant, are addressed or mitigated.
- A subcase is developed that supports the defeater by using the CAE Blocks to justify the claimed defeat. The subcase, because it is dealing with defeats, may also use a new disjunctive ("OR") block when a defeater is decomposed into different types or sources of defeat, where any one of these might be a defeater. Care must be taken if several levels of defeat defeating defeaters are used.

The way that defeaters are used graphically is shown in the short example in Section 3.3, and more details on background is provided in [4].

While we recommend that defeaters are used to attack a positive CAE case, supplementing existing practice, there is another style of case based on *Eliminative Argumentation* [6][7] which might also be considered. Here, rather than asking why a claim is true to identify subclaims (claim X is true *because* ...) it asks why it might be false (claim X is true *unless*....) and the case stops when all reasons of disbelief are identified and assessed. A justification is presented as a confidence map, which is rather like an assurance case with defeaters and hazards considered together: given the maturity of hazard analysis in the industry and potential confusion between hazards and defeaters, we prefer to keep these separate. Successful application of eliminative argumentation in assurance of real systems is reported in [8].

The recently updated Goal Structuring Notation (GSN) community standard introduces similar concepts to defeaters with its "Dialectic Extension" [5].

3.2 Defeaters identification

There is a variety of sources of experience and analyses that can be used as sources of defeaters. These can be grouped into three classes:

- hazard analysis and brainstorming
    - brainstorming particular topics and harvesting experience with cases and engineering justifications
    - supported by a form of preliminary hazard analysis using a guideword-based, Hazop-type approach applied to the justification process both generically and for specific cases

- knowledge of standards and relevant guidance
    - knowledge in standards and guidelines where lack of compliance can be a source of defeat
    - standards and guidance documents that explicitly consider vulnerabilities and things to avoid which can be directly considered as defeaters
- the literature on biases and argument fallacies and whether these are present in the case

3.2.1 Brainstorming and "hazard analysis"

Brainstorming is a useful technique for identifying defeaters. An example programme for a 60 minute defeater workshop is shown in Table 2.

Topics	Description	Approximate timing
Introduction to project and the workshop	Brief on project and methodology, and workshop approach.	10 mins
Silent brainstorm	Participants individually generate defeaters based on their experience of what goes wrong that undermines confidence in the case. These could be written on post-its, or shared through an online tool. Mainly silent - moderated, clarification questions.	10-15 mins

Topics	Description	Approximate timing
Discussion	Each participant introduces (some of) their "post-its". Participants briefly discuss each.	5-10 mins
Prompted review	Moderator prompts from the CAE case, selecting particular areas or claims to consider. Discussion around how the generic defeaters might manifest in this application area? What new issues might there be?	10-15 mins
Discussion	Classification and grouping of defeaters, and more elaboration. This may identify work to be done outside of workshop.	20 mins
Wrap up	Discuss immediate reflections, future work required, and any other next steps.	5 mins

Table 2: Example outline of a defeater workshop

The usual brainstorming rules should apply:

- Separate out sentencing and discussion from generation of ideas. This is meant to allow ideas to flow.
- Do some of the workshop in silence. "Silent brainstorming" is effective.

A collaboration tool such as GroupMap (https://www.groupmap.com/) can provide a canvas to position ideas and comment on them. An example of a canvas is shown below in Figure 2.

The four quadrants are

- generic issues (green) – defeaters from our experience and literature
- generic safety (yellow) – defeaters important to safety in general
- specific (oranges) – defeaters related to the specific systems being assessed but less safety relevant
- specific and safety (red) – defeaters important to safety related specifically to the specific systems being assessed

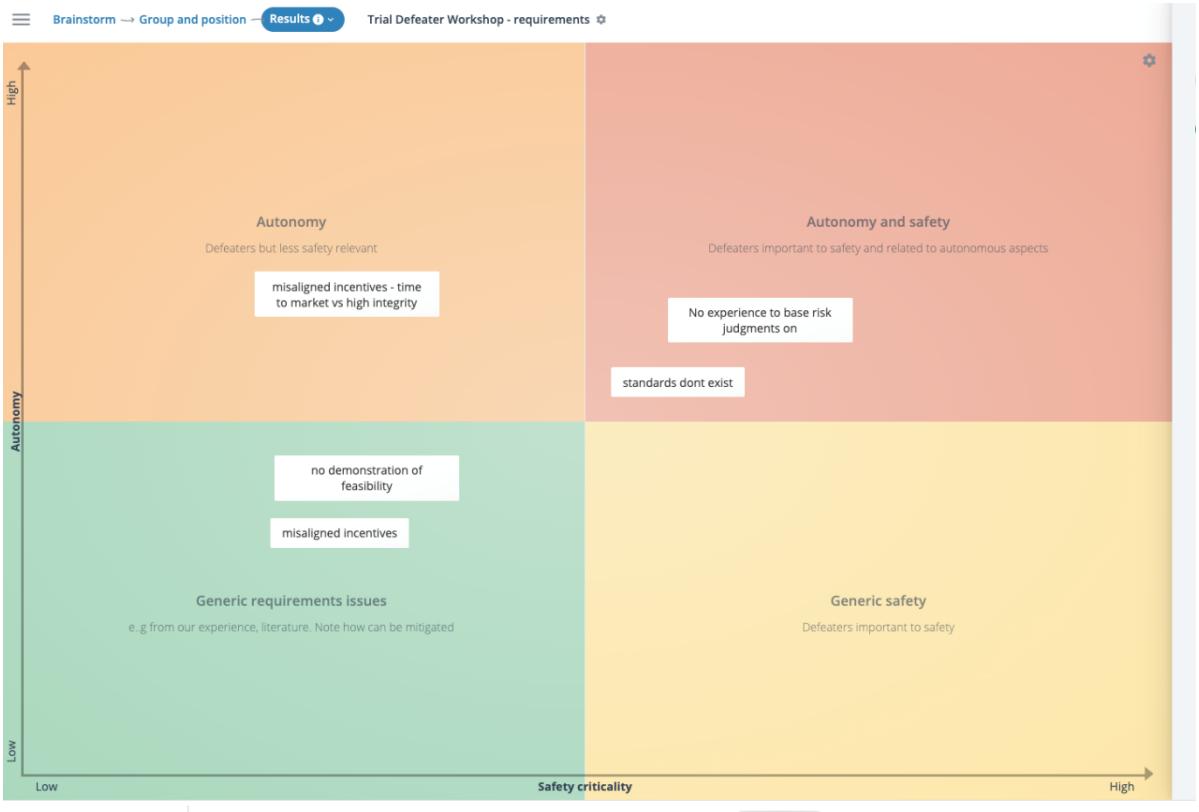

Figure 2: Example brainstorm canvas

Initially in the brainstorm, issues should be identified, and then as the workshop continues, move and detail them in the appropriate region of the grid according to whether they are generic requirement issues, generic safety issues, specific to the systems being assessed but not directly safety issues, or whether they involve both the specific system and safety.

Brainstorming workshops can provide a source of potential defeaters as identified by practitioners in the nuclear industry. Table 3 is an example containing defeaters relating to our understanding of a specific system.

Issue from brainstorming	Generic issues	Questions to probe for defeaters
Technical basis/understanding changes during production of case but is difficult to incorporate/accommodate	Understanding and changes to it not captured	What new knowledge has come to light during the case production that might impact it?

Issue from brainstorming	Generic issues	Questions to probe for defeaters
Asset and system changes over time not considered	Lack of understanding of the world Scope of case and how changes are dealt with	Does the case consider changes to the assets or objects it refers to? Is the scope of the case clearly defined with respect to change? Is there a need for an explicit "future" branch?
Scope creep if objectives or top claim poorly defined	Inadequate awareness of scope of case Case development process and review	Is the scope of the case clear? Are the objects (systems, components, environment) that the case refers to explicitly defined?
Not all aspects that support the case have been identified for example, hazards have been missed	Incompleteness	Are there implicit or explicit claims about completeness (e.g., of hazards)? Are these explicitly justified (e.g., via justifying side claims in CAE Blocks)?
Conservative assumptions skew output	Impact on decision of conservative assumptions	Are there conservative claims (e.g., bounds on reliability, worst-case timing)? What is the impact of these on the top-level claim and the decision that case supports?

Table 3: Brainstorming issues

One of the parameters we can vary in a brainstorm is the extent to which we try and drive the workshop systematically to make it effective and efficient without wanting to constrain. Following common practice from Hazop workshops we can use a guideword-based approach examining either generic arguments or the specific CAE of the system being considered. This approach can also be done in separate analysis activity.

Table 4 shows the questions identified for "arguments", during a preliminary defeater analysis using a guideword-based, Hazop type, approach.

Guide word	Interpretation	Weakness in case	Questions to probe for defeaters
Incomplete	Assumptions not clear, inference incomplete	Not valid but convincing Valid but not convincing	Do the arguments steps refer to accepted theories? Are the assumptions of these defined and addressed? Is the argument step explained (e.g., in narrative)? (Specific prompt trees like Figure 3 can be useful)

Guide word	Interpretation	Weakness in case	Questions to probe for defeaters
Missing	Extreme form of "incomplete"	Claims not supported by subclaims or evidence	Are there explicit argument steps between the evidence/subclaims and the claims they support? If arguments are omitted in the summary is this because it is a summary or does it reflect lack of rigour and understanding?
Too much	Inappropriate level of detail, irrelevant information	Cannot find what is wrong due to mental overload Not convinced	Is the argument step explained succinctly? Is all the information provided relevant to the argument?

Table 4: Hazard identification in Argument steps in a CAE case

We note that it might be judged incredible that missing arguments are not detected if using a graphical overview of the case with arguments required between claims. Instead, the issue is normally related to the quality of argument.

A similar approach can be developed for claims and evidence by adapting [3].

3.2.2 Use of guidance and standards

Standards and guidance documents can be used to identify vulnerabilities and properties of the system under assessment. Non-compliances or gaps with respect to relevant standards are potential defeaters, and part of an expert review for defeaters would be to assess their relevance and impact on the case. Indeed, the use of a behaviour-based case can provide a principled approach to justifying non-compliances.

For example, Annex II of the IAEA guidance [3] collects together vulnerabilities or weaknesses in nuclear C&I systems. It provides an example prompt tree, Figure 3, on the composition challenges to show that claims about the properties of individual system components are still valid when those components are combined within a particular context. The prompt tree suggests potential areas of difficulty, such as competition for resources, concurrency issues such as race conditions, and lack of robustness or completeness caused by failure to consider all possible modes of operation or ranges of input values.

Reviewing the case, we might identify there is a decomposition into different components. The safety case author or reviewer would start at the left-hand side of the prompt tree and ask increasingly more detailed questions about the case. For example, taking the competition for resources prompt, a defeater can be identified if the CAE structure has not addressed potential resource depletion when the individual components are combined to form a system. More detailed prompts are provided to explore the different types and sources of resource depletion.

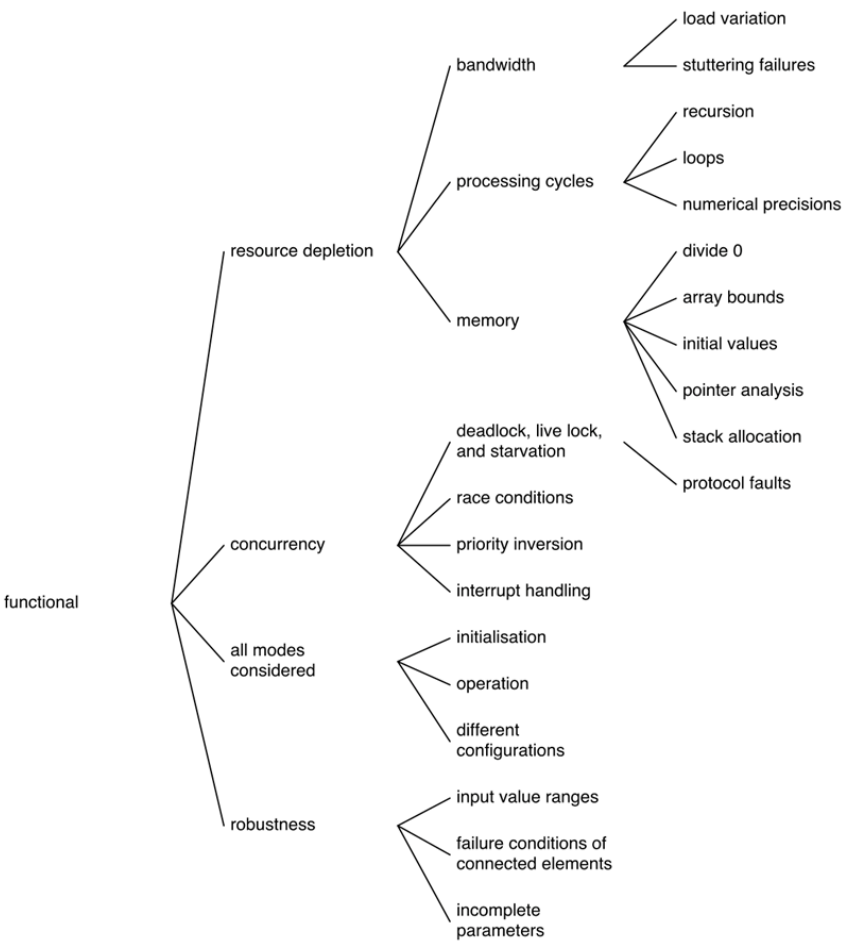

Figure 3: Example prompt tree from IAEA Guidance [3]

3.2.3 Fallacies

Another source of defeaters is the literature on argument fallacies. In previous projects we studied Greenwell's list of common types of safety case fallacies [9], and its impact on a CAE Block-based safety case. The fallacies can be grouped as follows:

Fallacies that are unlikely to occur. The requirement to provide side-claims (i.e., warrants) and supporting arguments for them if they are not obvious makes it unlikely that they will be relevant.

Fallacies that are difficult to make. The necessity of presenting the inference in CAE Blocks safety cases using a pre-defined set of analytical patterns makes it difficult (assuming proper application of the patterns) to make fallacies of composition or division.

Fallacies that are unlikely but still possible. The remaining fallacies are also much less likely to appear in CAE Block-based safety cases than in less formal representation of safety cases, but this does not prevent their appearance completely:

- appeal to improper authority
- fallacious use of language
- inductive generalisation
- red herring
- arguing from ignorance
- ignoring the counter-evidence
- gambler's fallacy
- arguing in a circle
- faulty analogy
- pseudo-precision

To avoid these fallacies the review process must address the proper representation of a safety case in CAE Blocks (to mitigate the fallacies from the first two groups) and define questions related to the third group of the fallacies listed above. As mentioned above, the CAE Blocks approach is specifically aimed at preventing many of these fallacies from occurring (e.g., normal form, blocks and guidance on use of language). We present some of the relevant common fallacies from the third group above in Table 5 below together with an analysis of how they might inform defeaters and mitigations.

Fallacy	Mitigations	Possible defeater indicators
Appeal to improper authority. This fallacy occurs when an argument supports a claim by referencing an authority that is in some way inappropriate or inadequate, such as individuals, committees or other groups, standard documents, "best practices," and system pedigree.	Making the argument steps explicit, specifically through the use of side claims to identify the inference being made. Evidence incorporation block forces author to justify and articulate rationale for selecting their evidence. Should help to reveal issues.	Citing authoritative standards which are not applicable. Appealing to research evidence which may not be relevant in practice. Misquoting evidence.
Fallacious use of language. When language used is ambiguous, or terms are used in different ways within a single justification (e.g. "system is safe", "all hazards are identified").	Concretion Block provides mechanism for making language use more precise. Checks that concretion is consistent across case. CAE normal form and claim, argument, evidence formulation guidance.	Imprecise or wrongly phrased claims, e.g., the claim does not make clear what the object and property of the claim are.

Fallacy	Mitigations	Possible defeater indicators
Inductive generalisation. Drawing a conclusion on the basis of a small / insufficient data set (e.g., claiming software correctness based on limited testing, or claiming true for a product line when only one prototype device assessed).	Identified by side claims e.g., in substitution of claim about a set of objects from claim of a single object. Explicit evidence integration and use of confirmation theory to identify weak evidence.	Claim does not distinguish between an object and a set of objects. Inadequate justification for generalizing e.g. from a device tested to one deployed. Insufficient evidence to support a claim – evidence incorporation not justified. Basis of the argument (the theory) not defined.
Red herring. When an argument includes sub-arguments that are entirely unrelated to the claim that the argument is intended to support, alongside sub-arguments that are related to the claim.	If a decomposition block is used, then the semantics are that all subclaims are needed to make the top-level claim true. Detect by review of the decomposition and use of guidance.	Unnecessary and confusing claims. Too many subclaims (e.g., rule of thumb that the number of subclaims should be fewer than 6). Overly complex case. Debris left in case from previous versions of the case.
Arguing from ignorance. An argument is made based solely on the lack of counter evidence (e.g., there have been no accidents in 10 years, therefore the system is safe).	Use of confirmation theory to identify weak evidence. Evidence incorporation block forces author to justify and articulate rationale for selecting their evidence. Should help to reveal issues.	False or misleading claims. Absence of credible evidence. Strength of evidence not assessed (e.g., with confirmation theory).
Ignoring the counter-evidence. Counter-evidence is ignored and the argument fails to acknowledge or consider that evidence, perhaps giving an impression that the evidence does not exist.	Evidence incorporation block forces author to justify and articulate rationale for selecting their evidence. Should help to reveal issues.	Generic indicators that this is an issue would be insufficient rigour and challenge. Lack of explicit counter claims or use of confirmation theory that would help identify counter evidence by exploring support for counter claims.

Fallacy	Mitigations	Possible defeater indicators
Confirmation bias The tendency to search for, interpret, favour, and recall information in a way that confirms one's pre-existing hypotheses or beliefs. Bias is a broad topic and addressed by mindset, process of review and challenge and making the justification accessible to many different stakeholders.	Blocks ask for justification of rationale. Review and challenge aim to remove or at least reveal bias.	Generic indicators that this is an issue would be insufficient rigour and challenge. Lack of confirmation theory to explore likelihood of evidence when the claim is false can help probe for bias. Reuse of large portions of another case without sufficient justification of relevance.

Table 5: Common fallacies and the associated symptoms

3.3 Applying and addressing defeaters

Defeaters can be identified at any stage of the Declare process as doubts, and should be added to the CAE structure as they are raised. Even at preliminary stages of formatting the top-level claim and argument in CAE, non-formalised defeater nodes can be added to the case. Formalised defeaters can only be connected to claim nodes, and they should be, ultimately, addressed or accepted.

3.3.1 Disjunctive defeater blocks

Decomposition blocks [16] are almost always interpreted as being conjunctive, i.e., all sub-claims must hold simultaneously for the top claim to be true. There are some cases, however, where the decomposition block can be interpreted as being disjunctive, i.e., any one of the sub-claims is sufficient to support or refute the top claim being decomposed [4]. These disjunctive blocks are primarily used in

- Generic templates of cases with alternative paths. In this situation, the assurance case is a generic template that may be instantiated in several ways according to specific properties or requirements. If, for example, there is a claim *P* that can be ensured either by sub-case *Q* or sub-case *R*, depending on how the system is implemented. In this case a disjunctive decomposition can be used and only the specific sub-case that is relevant to our system is developed.
- Negative cases requiring eliminative argumentation. While assurance cases are usually positive with the aim to construct and substantiate arguments in favour of the top-level claim, there are sometimes negative cases where the objective is to refute or challenge a claim rather than support it.

As defeaters challenge or refute a claim, a defeater disjunctive block can be used to represent multiple independent conditions or scenarios under which the claim could fail. This approach ensures all potential failure modes are systematically considered and addressed within the case.

Figure 4 below shows a simple example of using conjunctive and disjunctive decomposition blocks, and how defeaters are used with disjunctive blocks.

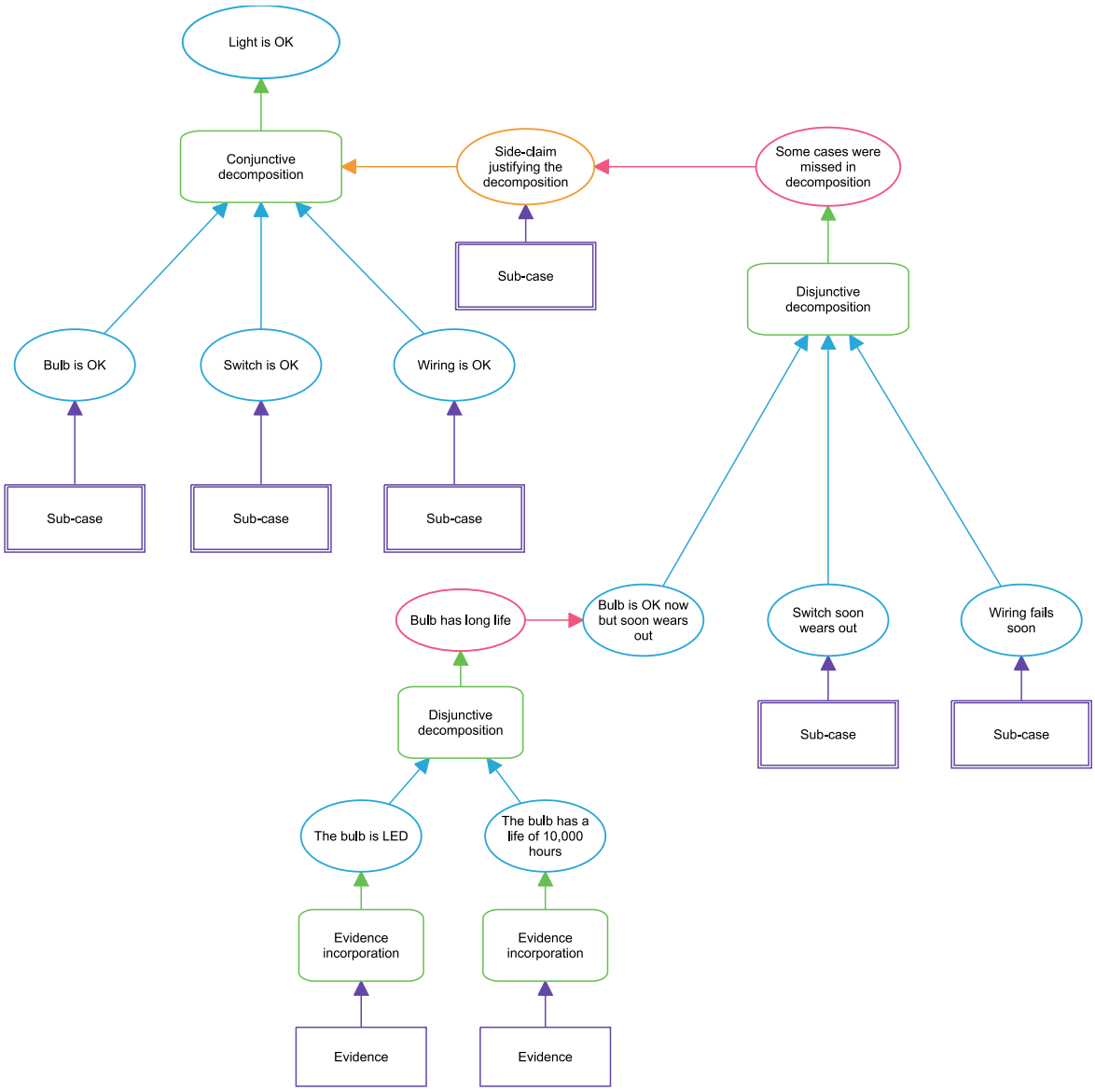

Figure 4: An example case showing conjunctive decomposition, and disjunctive decomposition with defeaters (adapted from [4])

The example is about verifying the functionality of a newly installed light. A positive assurance case is constructed to demonstrate that the light works correctly, with subclaims addressing the bulb, the switch, and the wiring. Each of these subclaims must be validated to ensure the top-level claim is true.

However, a defeater is introduced to challenge the claim by raising the possibility that some cases may have been missed in the decomposition. For example, even if the light works now, it might fail soon. This defeater is addressed using a disjunctive decomposition, where potential causes of failure (e.g., the bulb wearing out or being of poor quality) are considered independently. For the claim on bulb wearing out soon, a defeater is introduced that negates the failure claim. Evidence supporting the long life of the bulb is then

presented, which could include the bulb being an LED or having a warranty guaranteeing its durability. This disjunctive reasoning allows the defeater to be systematically refuted.

3.3.2 Addressing defeaters

Initial doubts and non-formalised defeaters in a case could be addressed by adding claims, arguments, evidence, and extending the narration of the case to account for identified gaps, omissions, etc.

To document the addressing of a formalised defeater in the assurance case, a second defeater addressing the initial defeater is added. This second defeater is followed by new claims, arguments and evidence explaining why and how the initial defeater has been addressed, along with the supporting narrative, which may also be required.

The structure of defeaters within a case can be exploratory or eliminative/exact. An example of an exact defeater in a real case is given in Figure 5, where the defeater negates the claim and proves it false. In this example, if the field data were collected in a different operating environment, then their analysis is not applicable to the intended operating environment.

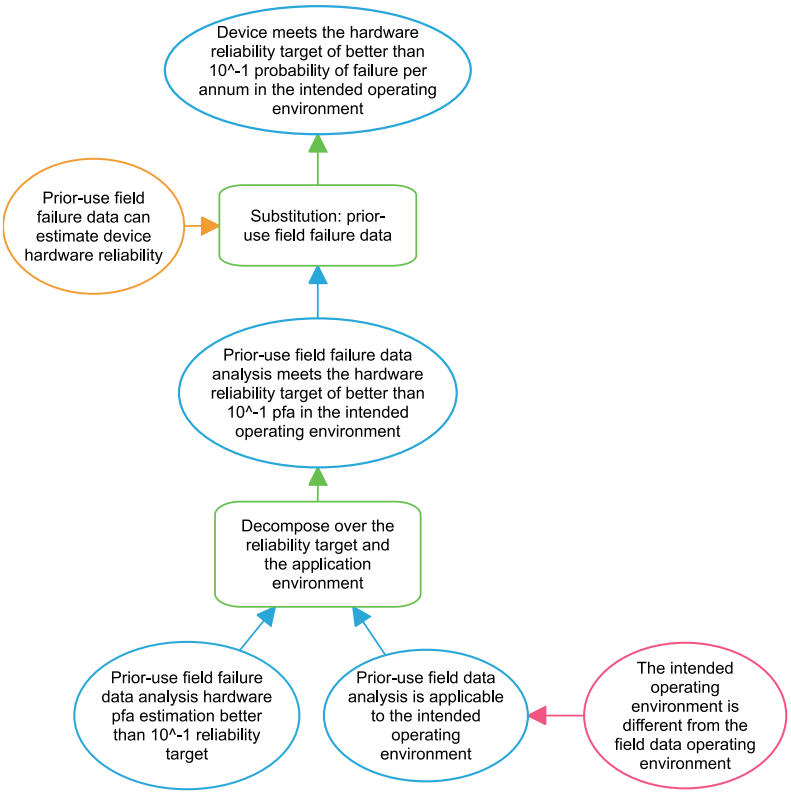

Figure 5: Exact/eliminative defeater example

One way to address the defeater is, for example, to provide evidence that the intended operating environment is more benign than the average operating environment when the field data was collected, and thus the data analysis is still applicable. We can express this graphically by marking the defeater as addressed (see the check decorator of the defeated node in Figure 6) and attaching a second defeater with evidence that defeats the initial defeater.

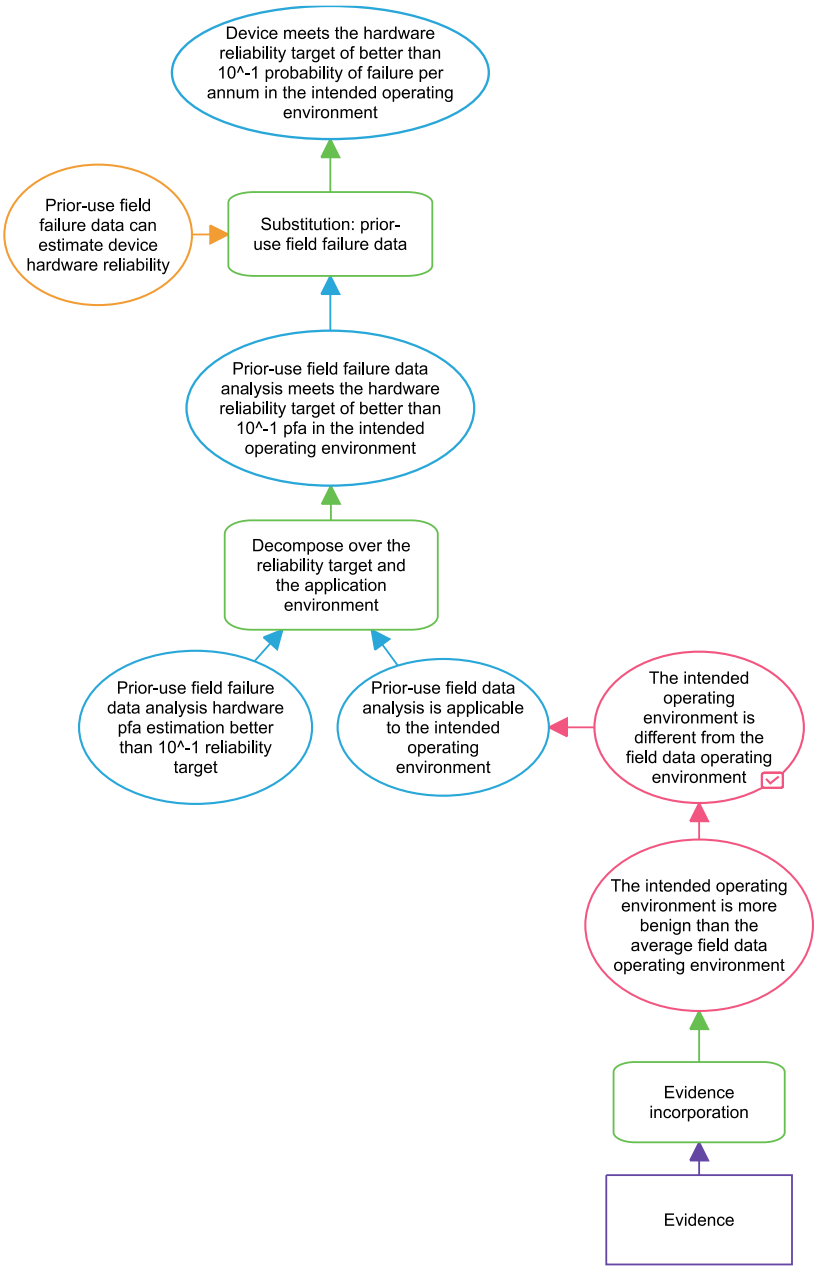

Figure 6: Addressed defeater

The different ways of addressing defeaters and expressing that in CAE form are subject to ongoing research.

4 Acknowledgements

This guidance draws on a number of sources developed in earlier Cinif, SSM and Adelard projects as well as published work by the authors.

5 Bibliography

[1] J. Pollock, "Defeasible Reasoning", in J E Adler & L J Rips (Eds.), Reasoning: Studies of Human Inference and its Foundations (pp. 451-470), Cambridge: Cambridge University Press, 2008.

[2] R. Bloomfield, J Rushby. "Assurance 2.0.", 2020.

[3] IAEA. Dependability Assessment of Software for Safety Instrumentation and Control Systems at Nuclear Power Plants. 2018.

[4] R. Bloomfield, K. Netkachova, J Rushby (SRI), Defeaters and Eliminative Argumentation In Assurance (to be published Jan 2024).

[5] Goal Structuring Notation Community Standard Version 3. The Assurance Case Working Group, York, UK, May 2021.

[6] John B. Goodenough, Charles B. Weinstock, and Ari Z. Klein. Eliminative induction: A basis for arguing system confidence. In Proceedings International Conference on Software Engineering, New Ideas and Emerging Results, pages 1161–1164, San Francisco, CA, May 2013.

[7] John B. Goodenough Charles B. Weinstock, Developing Sufficient (Justified) Confidence in Claims about a System, https://apps.dtic.mil/sti/trecms/pdf/AD1110340.pdf

[8] Simon Diemert and Jeff Joyce. Eliminative argumentation for arguing system safety—a practitioner's experience. In IEEE Systems Conference, 2020.

[9] W. Greenwell, C. Holloway, J. Knight, A Taxonomy of Fallacies in System Safety Arguments, Technical Report CS-2005-02, University of Virginia Department of Computer Science, 2004.

[10] Stanford Encyclopaedia of Philosophy, https://plato.stanford.edu/entries/reasoning-defeasible/

5.1 CAE main-guide

[11] R. Bloomfield, N. Chozos, Declare: CAE Main Guidance and Process, D/1284/43195/2 v1.0, April 2020.

5.2 CAE mini-guides

[12] R. Bloomfield, N. Chozos, P. Sketsiou, CAE Mini-Guide 1: One-Page Guidance, W/3601/43232/6 v2.0, December 2024.

[13] R. Bloomfield, N. Chozos, P. Sketsiou, CAE Mini-Guide 2: CAE Concepts, W/3602/43232/7 v2.0, December 2024.

[14] R. Bloomfield, N. Chozos, CAE Mini-Guide 3: Formulation of Top-Level Claim, W/3104/43195/1 v1.0, April 2020.

[15] R. Bloomfield, N. Chozos, P. Sketsiou, CAE Mini-Guide 4: Connection Rules, W/3604/43232/9 v2.0, December 2024.

[16] R. Bloomfield, N. Chozos, P. Sketsiou, CAE Mini-Guide 5: The CAE Building Blocks, W/3605/43232/10 v2.0, December 2024.

[17] R. Bloomfield, N. Chozos, CAE Mini-Guide 6: Summarising and Communication, W/3104/43195/1 v1.0, April 2020.

[18] R. Bloomfield, N. Chozos, P. Sketsiou, CAE Mini-Guide 7: Review and Challenge, W/3607/43232/12 v1.0, December 2024.

[19] R. Bloomfield, E. Karadotcheva, B. Phillips, P. Sketsiou, CAE Mini-Guide 8: Defeaters, W/3608/43232/13 v1.0, December 2024 (this document).