ADELARD

MINI-GUIDE 5: CAE BUILDING BLOCKS

Summary

This document is part of the Declare CAE guidance document set. It contains guidance on the use of the CAE Building Blocks.

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
Nick Chozos
Robin Bloomfield

Distribution
As per Cinif list in
Adelard D1284v10

Contents

Figures

Tables

1 Introduction

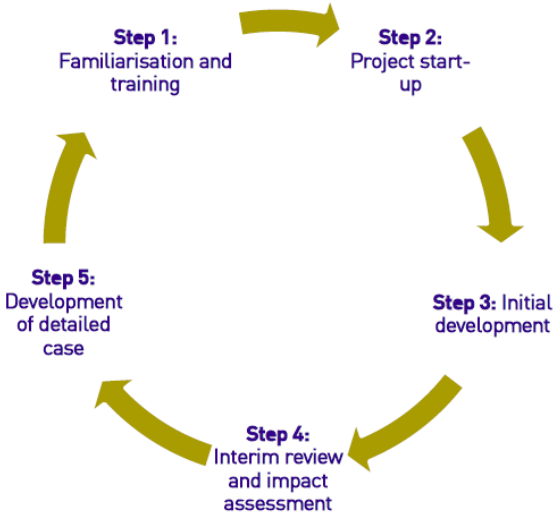
This document is a mini-guide that forms part of the Declare CAE guidance document set, and it provides guidance on the application of the CAE building blocks.

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.
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 5), and a full list of available mini-guides is given in Section 6.2.



	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

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 (CAE Building Blocks) applies throughout the CAE process.

CAE steps	CAE building blocks
Step 1: Familiarisation and training	Review and preparation.
Step 2: Project start-up	-
Step 3: Overview and initial development	Application during development of outline case – refinement is expected in next steps
Step 4: Interim review and safety case impact assessment	Full application on all nodes for purposes of review Several blocks will have been completed at this stage if used
Step 5: Development of a more detailed case	Iteration of Step 4 until all evidence has been linked to case Guidance may be used for review and challenge

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

The following section contains the detailed guidance on the use of CAE for the development of summary cases.

3 Mini-guide 5: CAE building blocks

This mini-guide contains a set of building blocks, i.e. fragments that are useful for expressing the safety justification. These can be used to decide which type of argument to apply for a specific type of claim, and guide the user through the process of elaborating that fragment in a careful manner, aiming at creating a complete and clear argument.

3.1 What are the CAE building blocks?

CAE building blocks are a series of archetypal CAE fragments that were derived from an empirical analysis of real cases in various domains, where we analysed what the cases were trying to express. They enhance the classical CAE approach with a standardised structure and an approach to how arguments are addressed. The five basic CAE building blocks are:

- Decomposition – this partitions some aspect of the claim in a "divide and conquer" approach
- Substitution – refines a claim about an object into another claim about an equivalent object
- Concretion – gives a more precise definition to some aspect of the claim
- Calculation or proof – used when some value of the claim can be computed or proved
- Evidence incorporation – incorporates evidence that directly supports the claim

The summary and the structure of these basic blocks are provided in Table 2. Additional information and guidance can be found in the paper [15].

CAE building blocks are based on the CAE normal form with further simplification and enhancements. The block structure contains enhancements in how arguments are addressed. We identify specific rules of the argument that are called "side-warrants". Side-warrants explain why we can deduce the top-level claim from the subclaims, and under what circumstances the argument is valid.

The side-warrant is in fact a type of claim and we may wish to challenge and demonstrate this for the specific case. We could do this either by justifying the side-warrant directly or by supporting the side-warrants with further subclaims and argument. We often refer to side-warrants as side claims.

The graphical scheme of a generic CAE block structure is shown in Figure 2 below. It shows *n* subclaims supporting an argument that justifies a top-level claim, with some of the key properties of the argument expressed as the side-warrant and supported by the system information and external backing.

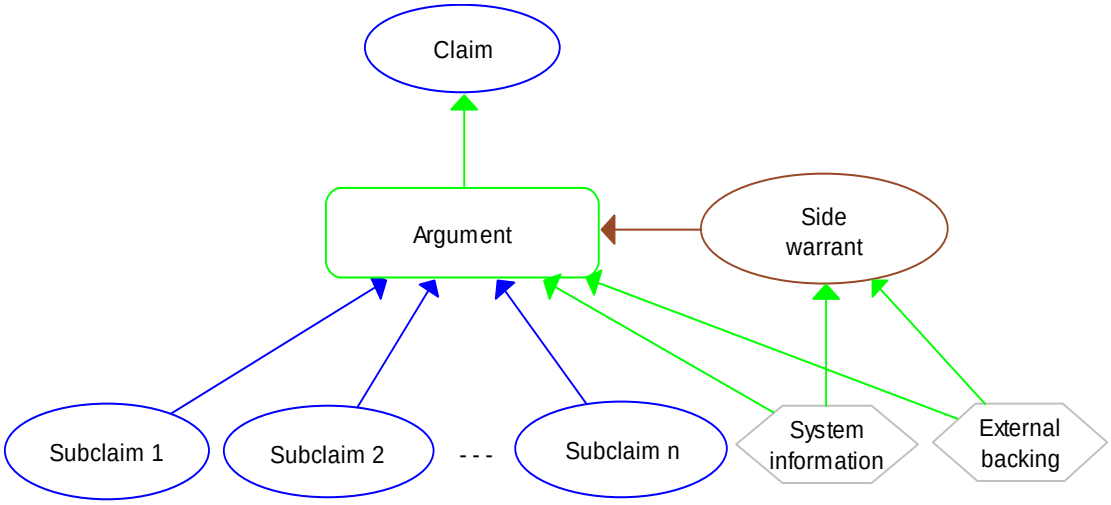
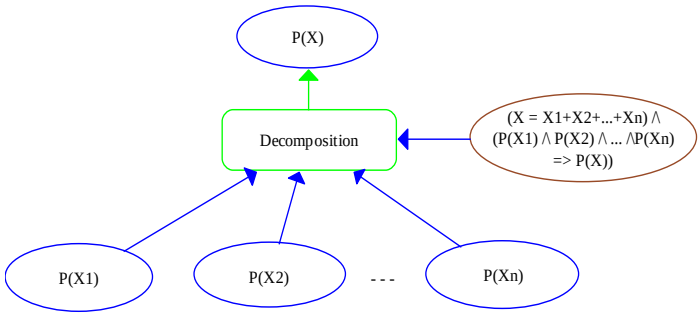
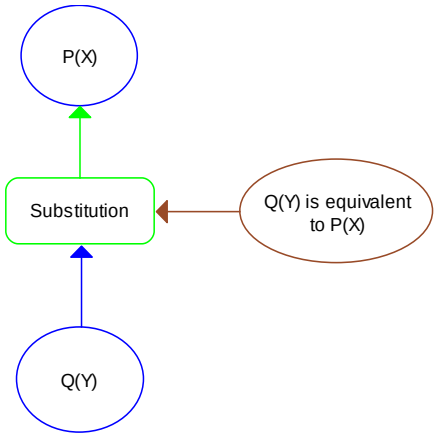
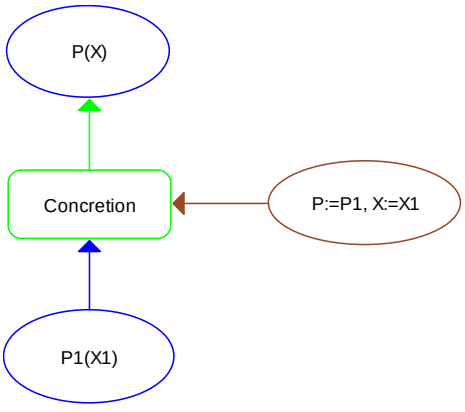


Figure 2: Generic CAE Block Structure

The overall justification for the block and its application can be included in the argument node narrative or accompanying text for the block. As shown in Figure 2 both the argument node and the side-warrant can be supported by additional data: system information and external backing. The former includes any system related information that drives the justification: models of system objects and properties, information from the product specification or user documentation, etc. The latter includes facts, guidance, theorems and theories that are appealed to as true statements of facts external to the warrant. The warrant might rely on external backing when demonstrating the claim, so if the backing itself is questionable, it must also be justified.

The side-warrant reminds us of the reasoning that we claim is true of the block. This may prompt us to detail the CAE structure further, as breaking it into more steps may identify how we can make the justification valid, or it may be that the block is a simple one (perhaps the top-level claim is just the conjunction of the subclaims). Alternatively it may be that we can appeal to some general result or authority to justify the side-warrant.

There are five basic CAE Building Blocks as shown in Table 2.

Structure	Description
	**Decomposition block** This block is used to claim that a conclusion about the whole object or property can be deduced from the claims or facts about constituent parts.
	**Substitution block** This block is used to claim that if a property holds for one object, then it holds for an equivalent object. Similarly, if a property holds for some object, then an equivalent property will also hold for the same object. The nature of the 'equivalence' will vary with the object and property and will need to be defined.
	**Concretion block** This block is used when a claim needs to be given a more precise definition or interpretation.

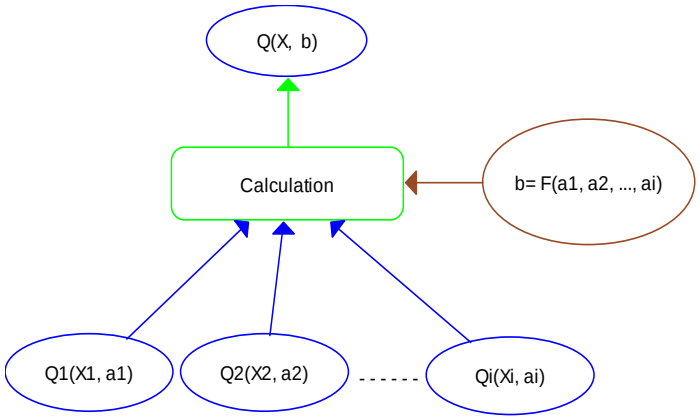
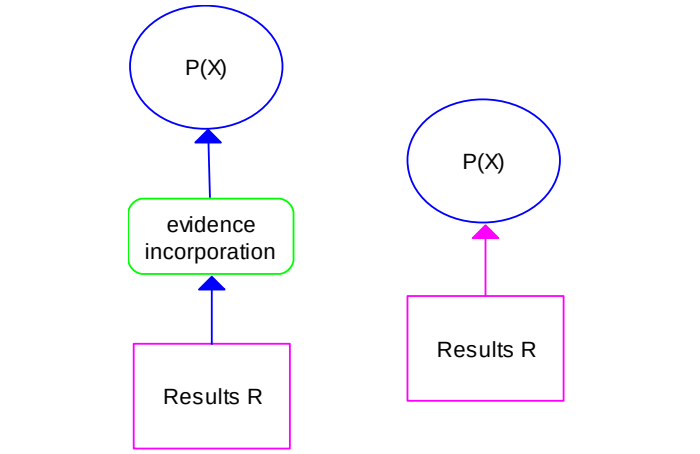
	**Calculation block** This block is used to claim that the value of a property of a system can be computed from the values of related properties of other objects. Show that the value *b* of property *Q(X, b, E, C)* of system *X* in environment *E* and confidence *C* can be calculated or proved from values $Q_1(X_1, a_1, E, C), Q_2(X_2, a_2, E, C), \ldots, Q_n(X_n, a_n, E, C)$
	**Evidence incorporation block** This block is used to incorporate evidence elements into the case. A typical application of this block is at the edge of a case tree where a claim is shown to be directly satisfied by its supporting evidence.

Table 2: Basic Building Blocks for Assurance Cases

3.2 How can the CAE blocks be used?

The CAE blocks are meant to support the creative process of constructing a case. They do not themselves show us how to architect cases, but provide a series of standardised ways of proceeding, either when a case is being developed top down or bottom up. The question that the case developer has to address is "which block could I use now?"

In order to support the teaching and deployment of CAE Building Blocks, we have created a visual guidance shown in Figure 3. We call it a "helping hand" as it is designed to help people structure assurance cases in an easier and more intuitive way by providing a "cheat sheet" on a hand with some hints and questions to answer. Instead of wondering what to do next and how to better expand the case, this approach shifts the question to an easier one: "which block is best to use?" and helps to find the answer by following the provided guidance.

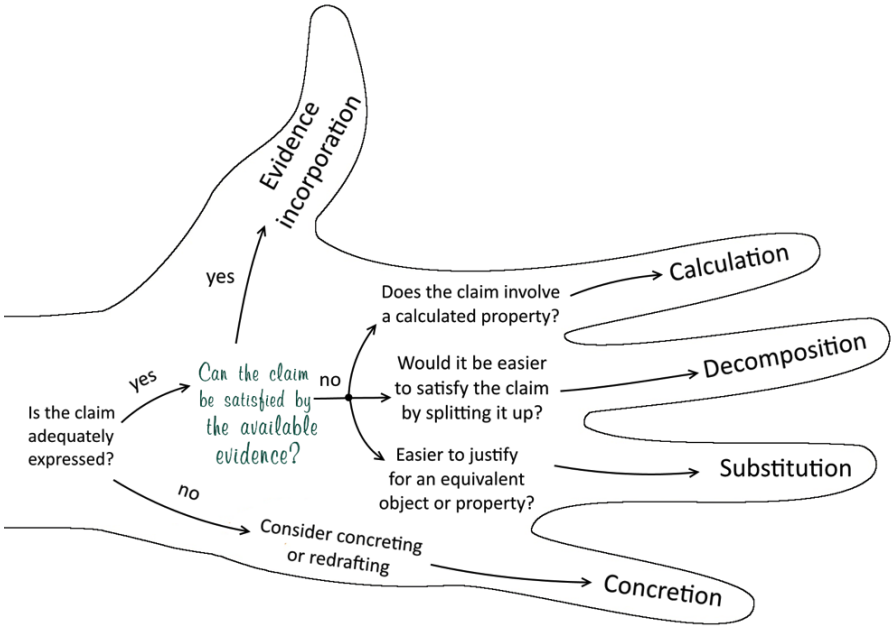


Figure 3: "Helping hand" – high level guidelines for selecting the CAE building block

The CAE blocks can be applied in three steps:

1. The first step involves selection of the block using the helping hand and its instantiation to the claims being considered. A rationale for selecting this block is given in narrative.
2. The next step involves adding the side-warrant that defines the *argument rule* being used.
3. The third step is to support the argument rule with narrative. Usually this requires further support either with a single evidence incorporation block or with a more detailed CAE structure.

The three steps are summarised in Figure 4 below.

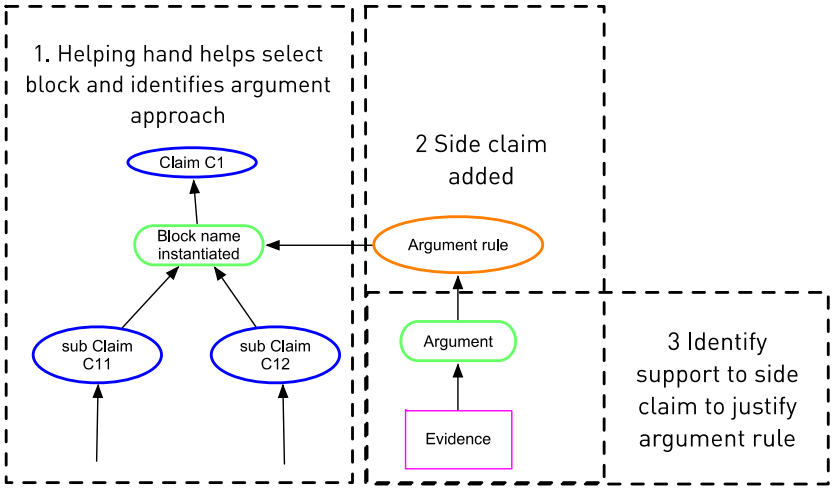


Figure 4: Summary of three steps of using CAE Block

Part of the skill in architecting a CAE fragment is in identifying the key properties that should be justified separately in the side-warrant. If we consider the simple example of Figure 5 we might find that the only side-warrant we can identify is *C11 /\ C12 => C1*, which makes the verification trivial (just modus ponens)[1] but pushes the justification into that for the warrant. Alternatively we may find that if a property referenced in the subclaims distributes then we can infer C1 from the subclaims.

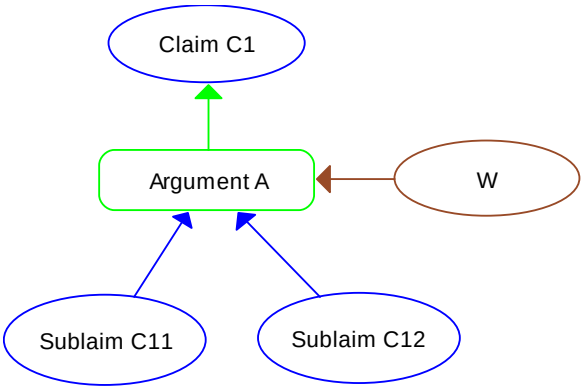


Figure 5: Simple example of CAE fragment with side-warrant

The side-warrant reminds us of the reasoning we are claiming is true of the block. This may prompt us to detail further the CAE structure as breaking it into more steps may identify how we can make the justification valid, or it may be that the block is a simple one (perhaps *C1* is just the conjunction of *C11* and *C12*). Alternatively it may be that we can appeal to some general result or authority to justify the side-warrant.

This guidance is based on [15].

3.3 The CAE "Decomposition block"

This section provides specific guidance on the application of the CAE Decomposition block. Experience has shown that the Decomposition approach is one of the most frequently used. This makes sense as in many cases the "divide and conquer" approach is obvious. Yet in making this practicable and rigorous there is a need to

- justify the reasoning that the combination of the parts do indeed make the whole
- be confident that there are gains in the effectiveness of the case of using the block (e.g. in efficiency of evidence production, in justification, in managing the case, in the communication of the case)

We first introduce the block and discuss the issues in applying it.

[1] We have experimented with a number of logical semantics for the CAE notation, a simple one comes from propositionalising the claims and the warrant so: *C11 /\ C12 /\ W => C1*

More specifically, we present how to instantiate the claim and side-warrant with the appropriate narrative, and how to develop the subclaim decomposition, as initially illustrated in Figure 4. We also discuss some specific examples where we expect this block to be used by providing further detail about its application.

3.4 The Decomposition block and its application

The Decomposition block is used to claim that a conclusion about the whole object or property can be deduced from the claims or facts about constituent parts and is typically structured as illustrated in Figure 6 below.

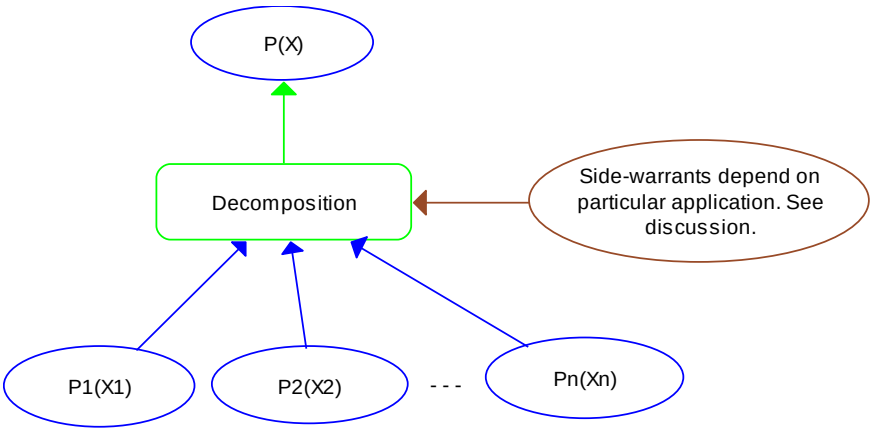


Figure 6: Decomposition block

With this very general block, useful side warrants depend on the specific application. However, if a less general decomposition is used then more powerful generic side-warrants can be defined e.g. if only the object is decomposed then we need to show that the property distributes over a composition operator; this is shown below in Figure 7.

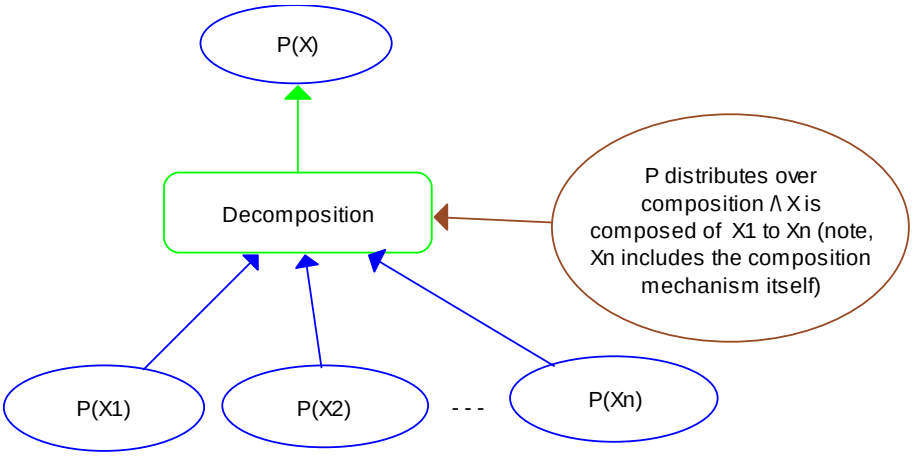


Figure 7: Example of single object decomposition

Experience has shown that deploying the Decomposition block makes sense as in many cases the advantages of a "divide and conquer" approach are obvious, e.g. for an argument based on compliance with

a standard, where the next level of decomposition is to consider each clause of the standard individually. By demonstrating that each of the standard clauses have been met, we can claim that the standard has been met. Other similar cases of decomposition might include meeting a requirement specification or addressing the hazards identified – demonstrating that a claim is met by the sum of the individual parts.

We often find that, when one takes this common approach, it is possible to omit a very important step – the provision of the rationale for taking this approach, and a justification for why this is adequate, in order to establish that the subsequent decomposition helps to justify the top level claim – is the sum of the constituent parts enough, or do we need to do more in order to achieve completeness? This can be more important in certain cases of decomposition than others; for instance, whilst it can be argued that a decomposition based on meeting each clause of a standard in order to demonstrate compliance with this standard can be obvious, this becomes more challenging when arguing that, e.g., a system is safe, by demonstrating that all its components are safe. What about the interaction between the system's components?

Even in the case of the standard clause-by-clause compliance it is important to note that the block structure requires that subclaims should all be true at once so that each claim has to be true given that all the other claims are true. To show that a system meets a standard by demonstrating each clause individually we would need to make sure that the means of compliance do not conflict. In addition, it would only be possible to show that all clauses are complied with if the standard is consistent and has no contradictions.

The need for each of the subclaims to be true together is also an issue when considering bottom up development of a claim. Confidence in individual subclaims might be demonstrated in one environment (context) and then would need to be shown in an environment where the other subclaims were also satisfied. This might need additional subclaims about the independence of claims or interactions between them.

There is also the issue of ensuring all the evidence is considered. For example, there might be claim about high reliability supported by evidence of a long time between failures. Yet these failures might contradict another claim about functional correctness.

Developing and trying to apply the side-warrant that identifies the argument rules being used provides a systematic way of identifying the additional aspects that need to be considered.

In addition, some of the rationale for the decomposition approach may often be required further up in the overall argument structure. For instance, a decomposition for claims such as "all hazards are controlled" or "all requirements are met" would require further up in the justification a clarification of what "all" means, and an explanation of how they have been identified. The top-level claim for the decomposition block then becomes a claim about "all identified" or "all specified" objects or properties and so the individual elements in the decomposition are defined.

In summary, the formal definition of the decomposition block includes a number of powerful features - the conjunction of subclaims, the definition of the side-warrant – that help ensure that it is both correct and valid but the implication of these need elaborating to support application of the block. To do this we define an application process.

3.5 Workflow

As discussed earlier and in Figure 4, the first step of the process is the selection of the candidate block. As indicated in Figure 3 and using the "CAE helping hand", at this point, we would have identified that the decomposition might be a good approach.

The workflow is shown schematically below in Figure 8.

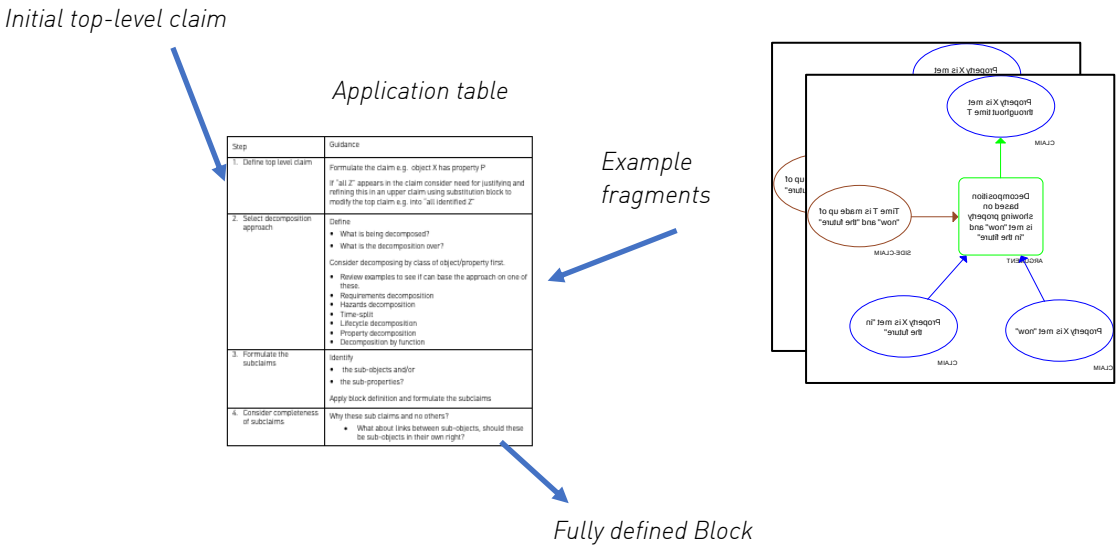


Figure 8: Block application workflow

A workflow in using the block top-down is supported by the application guidance in Table 3 that has the following steps:

1. Identify / define top level claim
2. Select candidate decomposition approach (some starting mini-examples are provided)
3. Apply block definition - define subclaims
4. Consider completeness of subclaims
5. Assess how the subclaims might be supported
6. Experiment with side-warrants – argument rules
7. Assess validity of side-warrant rules including identifying rebuttals
8. Assess benefits of decomposition on the next stages of the case and on evidence generation
9. Assess how best to represent the decomposition i.e. what mixture of graphical and tabular approaches

Each stage may be iterated and at each stage you may decide that the approach is not the best one or is not viable and seek alternative ways of structuring the case e.g. by returning to the helping hand.

To support these steps we provide guidance on the generic application of the Decomposition Block and provide short examples of the following types of decomposition:

- Requirements decomposition
- Hazards decomposition

- Time-split
- Lifecycle decomposition
- Property decomposition
- Decomposition by function

These examples are fragments that can be used to support the use of the Decomposition Block: the particular application should consider the generic application tables in order to define the detail that is important on the specific application being done. Appendix A provides an example of using the table for a requirements decomposition.

Step	Guidance
1. Identify / define top level claim	Formulate the claim e.g. object X has property P If "all Z" appears in the claim consider need for justifying and refining this in an upper claim using substitution block to modify the top claim e.g. into "all identified Z"
2. Select decomposition approach	Define • What is being decomposed? • What is the decomposition over? Consider decomposing by class of object/property first. Review examples to see if can base the approach on one of these • Requirements decomposition • Hazards decomposition • Time-split • Lifecycle decomposition • Property decomposition • Decomposition by function
3. Apply block definition - define subclaims	Identify • the sub-objects and/or • the sub-properties Apply block definition and formulate the subclaims

Step	Guidance
4. Consider completeness of subclaims	Why these sub claims and no others? • What about links between sub-objects, should these be sub-objects in their own right? • What else is in the object? What else is it connected to? • Does the object explicitly include environment and configuration, if not is this justified? And/or • What about interactions between sub-properties, should these be subclaims in their own right? • Is it clear what object the property applies to?
5. Assess how the subclaims might be supported	• Are the subclaims more easily demonstrated than the top-level claim? • Assess viability of evidence collection to support subclaims • Does evidence address multiple subclaims? If so, should the claims be merged together into different subclaims or should the evidence be more structured to delineate which subclaim it supports? • Can evidence generation be made more efficient by making claims more independent e.g. by additional claims? • Would it be better to use another decomposition block first e.g. decompose by architecture or dependability attribute, or decompose by stakeholder?
6. Experiment with side warrants – argument rules	How does the generic side warrant apply? Can we identify a more specific side-warrant to make the argument more precise and easier to justify? • What are the specific rules being applied to show that the subclaims support the top-level claim? • Can these rules be improved to allow for more independent demonstration of the subclaims?
7. Assess validity of side-warrant rules including identifying rebuttals	Can we validate the side-warrant? Are we dealing with an object or property that can be decomposed? (e.g. is the property, like beauty, holistic and not easily decomposed) Can we find a valid rule to define the side warrant? Does the side-warrant need to be developed further into supporting claims, arguments and evidence? Can you identify rebuttals or issues that would invalidate the reasoning? Should these become negated subclaims? Add subclaims or document issues to be resolved before using approach

Step	Guidance
8. Assess benefits of decomposition on the next stages of the case and on evidence generation	What are the costs and benefits of decomposition? • Assess impact of decomposition on further development of the case e.g. on evidence generation • Assess impact on communication of the case
9. Assess how best to represent the decomposition i.e. what mixture of graphical and tabular approaches	Use graphical approach when less than 10 subclaims (typically more than 6 suggests a reworking) Use a tabular approach for more fine-grained decomposition. (See Figure 9 – this figure is a screenshot from the ASCE tool)

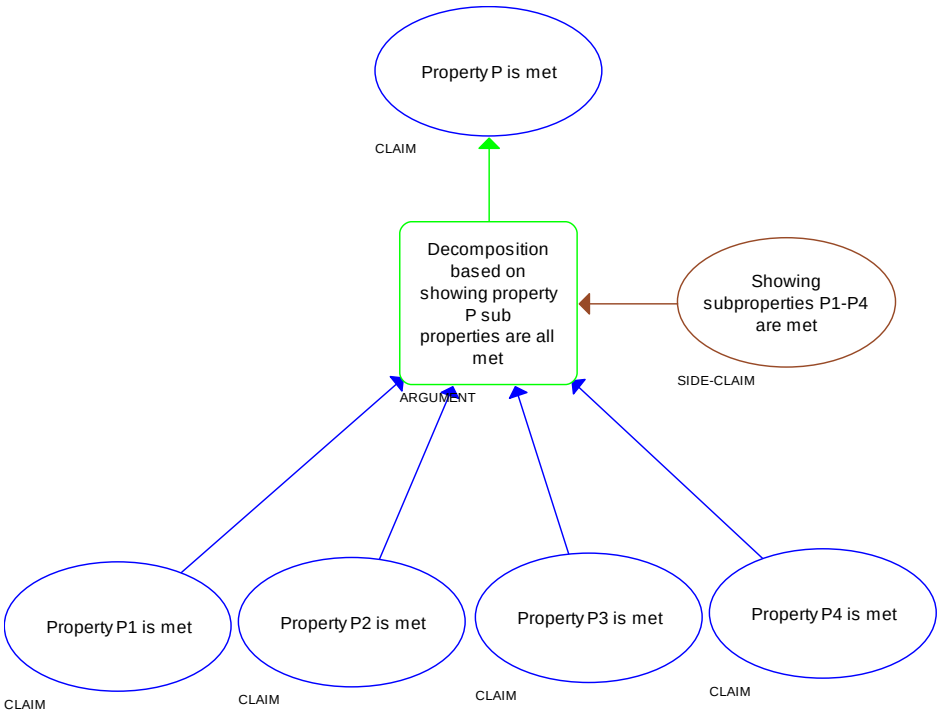
Table 3: Generic decomposition application guidance

3.6 Example starting points

To support the application of the Decomposition block we provide some summary examples. These should be used as part of the overall process of applying the Block (and not taken as finished examples).

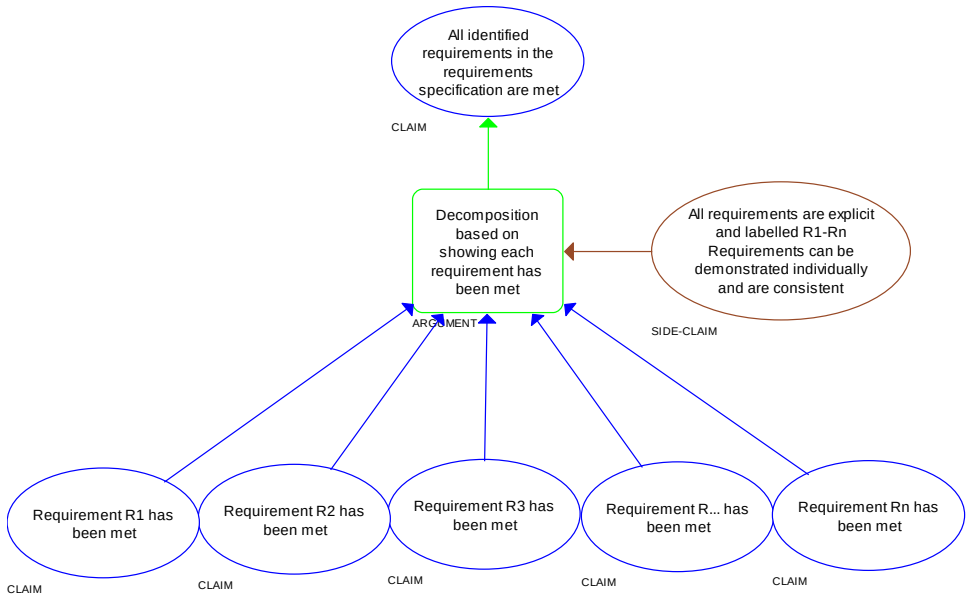
3.6.1 Property decomposition

Top-level claim	Property P is met
Summary	Decomposition is based on showing that Property P is decomposed into sub-properties P1, P2, P3, P4 e.g. security into integrity, availability and confidentiality

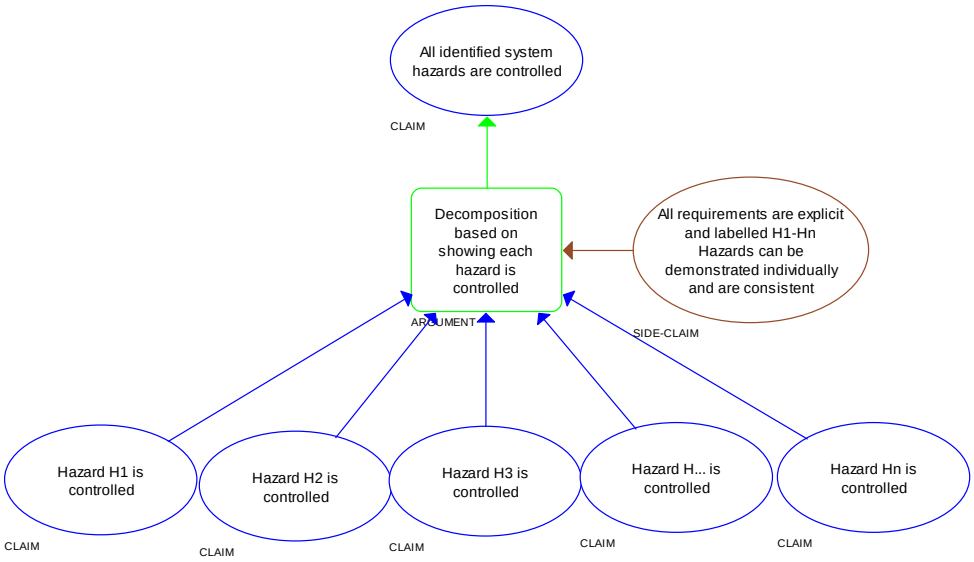
Diagram	
Decomposition approach	This is commonly done in cases where a certain property is made up of other sub-properties. For instance, dependability is typically seen as an "umbrella" term, made up of safety, security, integrity, fail-safety, maintainability etc. If this is the desired approach, then the next step in the decomposition will be to address each of these sub-properties one by one.
Application issues	Consider if the diagram should make explicit which system has that the property. The side-warrant provides a justification that showing the System/object has all the sub-properties then it has the overall property of the claim. In order to develop this argument appropriately, like all arguments, it is important to provide a basis for this approach: for instance, a technical / academic publication with definitions of these properties and their terminology will help to gain confidence that the decomposition is complete and consistent. The diagram shows four properties, it can be easily generalised to other numbers.

3.6.2 Requirements decomposition

Top-level claim	All identified requirements in the requirements specification are met
Summary	Decomposition based on demonstrating that each of the requirements in the specification is met

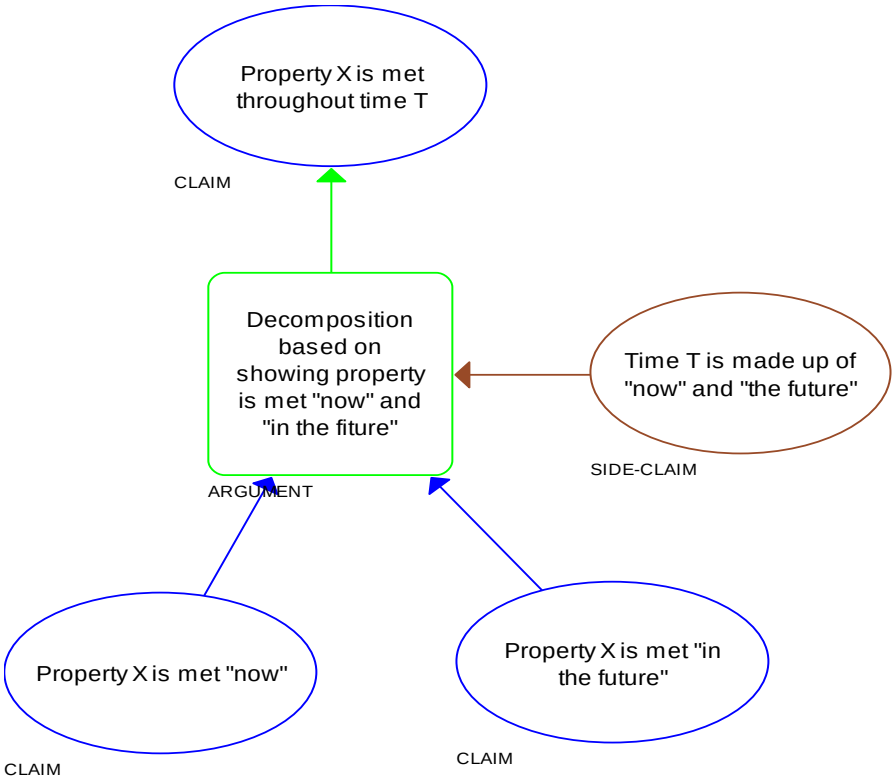
Diagram	
Decomposition approach	Typically, requirements are recorded in a requirements specification. You may opt to re-present each of the requirements as individual subclaims. In this situation it may be useful first to decompose over different *classes* of requirements as these might be mapped to different technical approaches (e.g. verification of logical properties, testing of robustness) or to different parts of the architecture and then consider individual requirements. There is often little to be gained by showing each requirement separately if they are subsequently dealt with in a similar manner. In terms of presentation a combination of graphical and tabular approaches could be used. A subclaim could refer to a traceability matrix that helps the reviewer to trace each of the requirements through to V&V evidence, in order to ascertain that all requirements have been met.
Application issues	In justifying the block we will need to describe how the requirements are combined together and in developing the side-claim, why the combination of all these requirements meets the top level requirement. The interaction between requirements needs to be addressed • to show that it is worth addressing each requirement individually • to include a claim about "interaction of requirements addressed" A possible rebuttal is inconsistency of requirements so this might be added as negated side claim to support the approach and can be explored early in the project.

3.6.3 Hazards decomposition

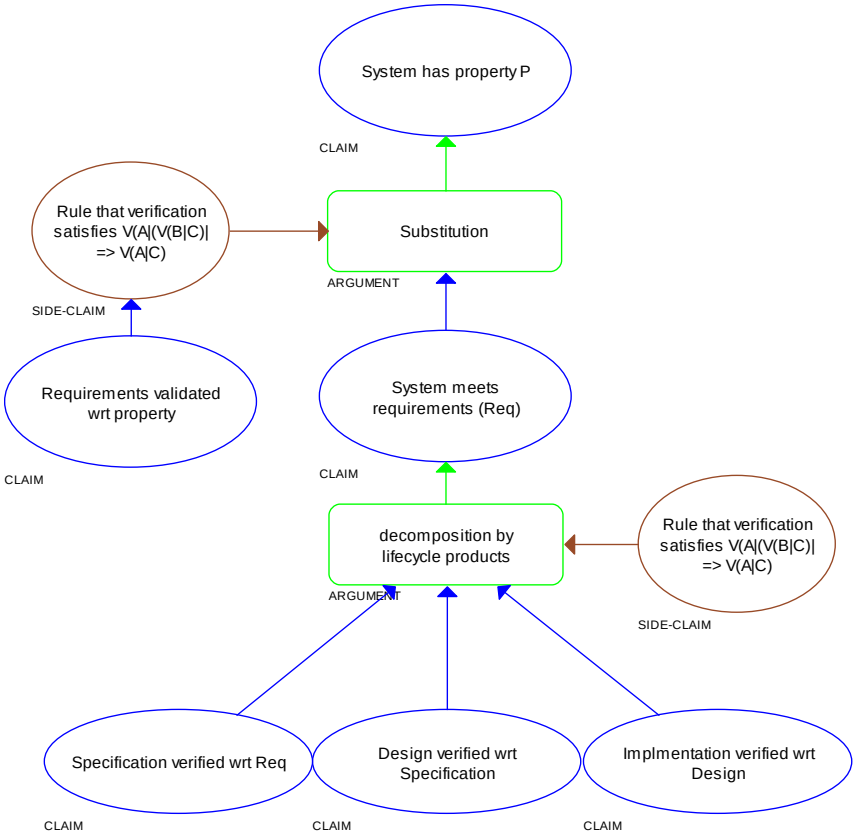
Top-level claim	All identified system hazards are controlled
Summary	Decomposition based on demonstrating that each of the hazards identified is controlled
Diagram	

Decomposition approach	Similarly with the requirements-based example discussed above, with a hazard decomposition, we would expect to see subclaims addressing each class of hazard. E.g. the USA Food and Drug Administration (FDA) requires that the following classes of hazard are addressed for medical devices: Operational, Hardware, Mechanical, Software, Use, Chemical and Biological. After decomposition by class of hazards, one could then avoid the repetition by listing each hazard if the content has already been recorded in a hazard log.
Application issues	The claim concerns "*identified* hazards" and avoids some of the completeness issues in using the Block but there will be a proceeding claim that has to address the issue of whether they are identified sufficiently. The interaction between hazards and controls needs to be addressed to show that it is possible to address each hazard or hazard class individually. If hazards are addressed by hazard class, the completeness of the hazard classes needs justifying e.g. from regulatory guidance, standards. The approach for dealing with hazards if there is more than one class needs to be defined. There is a need to ensure that the system and environment context are sufficiently explicitly defined as the control and impact of hazards will be very dependent on these.

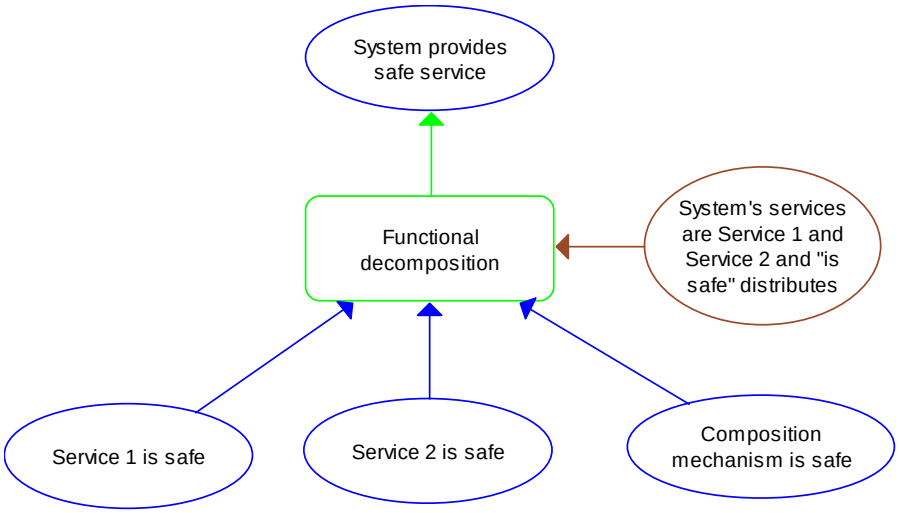
3.6.4 Time-split

Top-level claim	System meets Property X over time T
Summary	Decomposition based on demonstrating that a system property is met over a time period – typically this is done by considering the property being met "now" and "in the future".
Diagram	
Decomposition approach	In this case, we are claiming that a property for a system/component is met – typically now, and in the future.
Application issues	The meaning of "now" and "future" will need definition. One approach is to consider "now" to mean before any of specified set of events to occur and the "future" to be defined by a set of events such as failure of components, maintenance actions, upgrading. There may be a need to widen the definition of "system" to deal with the future events e.g. incident reporting mechanisms, maintenance organisations. The "now" branch of the case might include design features to support the handling of the future events and hence have links to the claims about the future.

3.6.5 Lifecycle decomposition

Top-level claim	System has property P
Summary	Decomposition based on considering each of the phases of the lifecycle. This could be either through-life (e.g. design to disposal) or a development lifecycle.
Diagram	
Decomposition approach	The approach used is that to show a system has a property we show that the requirements define this property and this is preserved as the system is implemented: the specification is verified with regards to the requirements, the design is verified with regards to the specification and so on.
Application issues	The side warrant needs to explain how the verification establishes that the implementation will meet the required property. There are issues of derived properties and of non-functional properties leading to functional properties as the lifecycle progresses. There may be aspects of the property that are only shown at an implementation level.

3.6.6 Decomposition by function

Top-level claim	System provides safe service
Summary	Decomposition based on showing that all functions of a system meet a certain property (e.g. are "safe")
Diagram	
Decomposition approach	A break-down based on demonstrating each function of the system. In this example there are two services and a mechanism for combining them.
Application issues	The side-warrant in this example will need careful justification, as it is not necessarily the case that safety properties can be split in this manner. In particular, it will be necessary to show that the individual functions are safe in combination and not just in isolation. For example, imagine heaters to regulate the temperature and blowers or valves to regulate the pressure. Doing both at the same time could cause bigger changes than anticipated.

4 Summary guidance

- CAE building blocks are fragments that are useful for expressing the safety justification. These can be used to decide which type of argument to apply for a specific type of claim, and guide the user through the process of elaborating that fragment in a careful manner, aiming at creating a complete and clear argument.
- There are five types of blocks: Decomposition, Substitution, Evidence Incorporation, Concretion, Calculation
- This guidance contains a "helping hand" to help the users decide which block to use
- This guidance contains advice on "starting points" for different types of decomposition and a worked up example (Appendix A)

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.

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

6.3 Other

[9] P.G. Bishop and R.E. Bloomfield, The SHIP Safety Case - A Combination of System and Software Methods, in SRSS95, Proc. 14th IFAC Conf. on Safety and Reliability of Software-based Systems, Brugge, Belgium, 12-15 September 1995.

[10] P.G. Bishop and R.E. Bloomfield, MJP Van der Meulen, "Public domain case study: An example application of the CEMSIS guidance", v1.0, 26/3/2004, WP5 Deliverable, http://www.cemsis.org

[11] P.G. Bishop and R.E. Bloomfield, "A Methodology for Safety Case Development", Safety-critical Systems Symposium 98, Birmingham, UK, Feb 1998, ISBN 3-540-76189-6.

[12] T. Kelly. "The goal structuring notation–a safety argument notation", Proc. DSN 2004 Workshop on Assurance Cases, 2004.

[13] ISO/IEC 15026-2:2011, Systems and software engineering — Systems and software assurance, Part 2: Assurance case.

[14] Adelard Safety Case Development Manual v1.1, Adelard, http://www.adelard.com/resources/ascad/

[15] R. Bloomfield and K. Netkachova. "Building blocks for assurance cases", S 2014 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW),3-6 Nov. 2014 doi: 10.1109/ISSREW.2014.72.

[16] K. Netkachova, O. Netkachov, R. Bloomfield. "Tool Support for Assurance Case Building Blocks, Providing a Helping Hand with CAE", Lecture Notes in Computer Science, Computer Safety Reliability and Security, pp 62-71, doi: 10.1007/978-3-319-24249-1_6

[17] R. Bloomfield, "Deploying Claims Arguments and Evidence (Declare) Part 2: Guidance on construction and review". Adelard document reference D/927/43131/3, issue v1.0, December 2015.

[18] C. Menon, R.E. Bloomfield, T. Clement, "Interpreting ALARP", in Proceedings of the 8th IET Systems Safety Conference, 2013.

[19] R. Bloomfield, N. Chozos, S. Guerra, K. Netkachova, Guidance on CAE: Concepts, blocks, templates. Adelard document reference W/2252/140001/10, 2014.

[20] R. Bloomfield, N. Chozos. Declare2: Preliminary guidance to support the use of CAE. Adelard document reference D/1038/43136/2, issue v10. December 2016.

[21] R. Bloomfield, N. Chozos. Declare3: Guidance on CAE concepts. Adelard document reference D/1061/43150/1, issue v0.2, March 2017.

Appendix A
Application of decomposition guidance

This section contains an example illustrating how the decomposition block may be applied. More specifically, we have applied the application guidance table to requirements satisfaction.

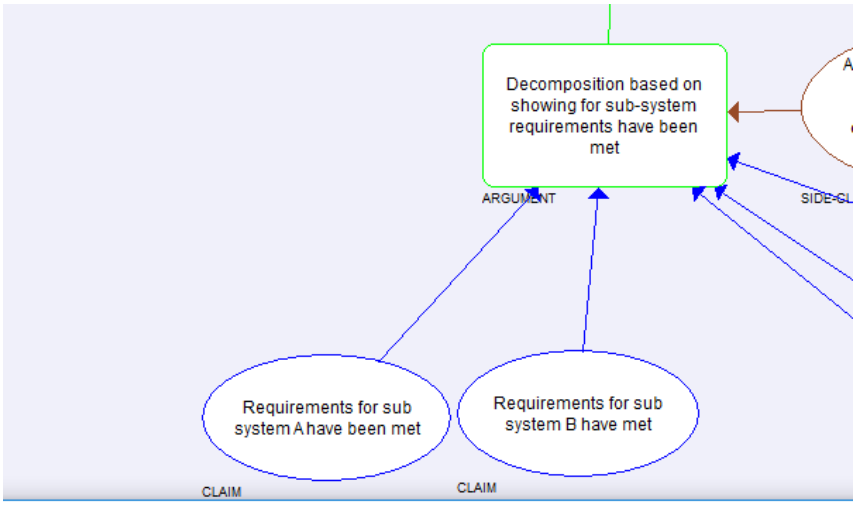
Step	Generic Guidance	Application
1. Define top-level claim	Formulate the claim e.g. object X has property P. If "all Z" appears in the claim consider need for justifying and refining this in an upper claim using substitution block to modify the top claim e.g. into "all identified Z".	Original top-level claim: all requirements are met. Using substitution block modify the top claim to: • All requirements identified in the requirements specification are met (Typically, requirements are recorded in a requirements specification.)
2. Select decomposition approach	Define • What is being decomposed? • What is the decomposition over? Consider decomposing by class of object/property first.	Decomposed by individually identified requirement. Decomposition over requirements numbering and classification system e.g R.n.m. Consider decomposing by *class* of requirements first. It may be useful first to decompose over different *classes* of requirements as these might be mapped to different technical approaches (e.g. verification of logical properties, testing of robustness) or to different parts of the architecture (e.g. by sub-system) and then consider individual requirements. There is often little to be gained by showing each requirement separately if they are subsequently dealt with in a similar manner.
3. Formulate the subclaims	Identify • the sub-objects and/or • the sub-properties? Apply block definition and formulate the subclaims	The sub-objects are "Requirement R.n.m". Property P "is met" is not changed. Subclaims of form "Requirement R.n.m is met".

Step	Generic Guidance	Application
4. Consider completeness of subclaims	Why these subclaims and no others? • What about links between sub-objects, should these be sub-objects in their own right? • What else is in the object? What else is it connected to? • Does the object explicitly include environment and configuration, if not is this justified? And/or • What about interactions between sub-properties, should these be subclaims in their own right? • Is it clear what object the property applies to?	Completeness is provided by enumerating over all the identified requirements in the specification: the top-level claim is about *identified* requirements. In this example we assume that the requirements are meaningful individually and that version, and configuration are well defined. (Completeness is also addressed in considering the side-warrant in step 7)

Step	Generic Guidance	Application
5. Assess how the subclaims might be supported	Assess viability of evidence collection to support subclaims. Are the subclaims more easily demonstrated than the top-level claim? Does evidence address multiple subclaims? If so, should the claims be merged together into different subclaims or should the evidence be more structured to delineate which subclaim it supports? Can evidence generation be made more efficient by making claims more independent e.g. by additional claims. Would it be better to use another decomposition block first e.g. decompose by architecture or property? Or decompose by stakeholder?	Consider whether evidence will address multiple requirements. Consider if these should be collected together into different subclaims. Consider grouping of requirements. Would it be better to decompose by architecture or property? Would it be better to group requirements by stakeholder?
6. Experiment with side warrants – argument rules	How does the generic side warrant apply? Can we identify a more specific side-warrant to make the argument more precise and easier to justify? • What are the specific rules being applied to show that the subclaims support the top-level claim? • Can these rules be improved to allow for more independent demonstration of the subclaims?	From the generic side warrant rule need to show that if "is met" can be shown for each requirement individually then it applies to the set of requirements. In general • there may be requirements that reference other requirements • there may also be requirements about requirements In this example we assume that the requirements are meaningful individually. Need a claim that the Requirements Specification only has R.n.m with n and m given a range.

Step	Generic Guidance	Application
7. Assess validity of side-warrant rules including identifying rebuttals	Can we validate the side-warrant? Are we dealing with an object or property that can be decomposed? (e.g. is the property, like beauty, holistic and not easily decomposed.) Can we find a valid rule to define the side warrant? Does the side-warrant need to be developed further into supporting claims, arguments and evidence? Can you identify rebuttals or issues that would invalidate the reasoning? Should these become negated subclaims? Add subclaims or document issues to be resolved before using approach.	Review Requirements Specification to see if side warrant valid i.e. all requirements have been uniquely defined in the specification. Either develop a subclaim "no implicit, referenced out or unnumbered requirements" or add to meaning of "identified" in the derivation of the top level claim for the Block. Requirements also need to be well defined in order to be met. Add subclaims to side warrant about these aspects. These conditions could become a rebuttal: assess early on to reduce risk of case not being feasible.
8. Assess how best to represent the decomposition i.e. what mixture of graphical and tabular approaches	Use graphical approach when less than 10 subclaims (typically more than 6 suggests a reworking). Use a tabular approach for more fine grained decomposition. (See Figure 9 for an example – this example was developed using the ASCE tool)	Use graphical approach for each class of requirements and then a tabular approach. Subclaims could refer to a traceability matrix that helps the reviewer to trace each of the requirements through to V&V evidence, in order to ascertain that all requirements have been met. (See Figure 9 for an example.)
9. Assess if the decomposition adds value to the case	What are the costs and benefits of decomposition? • Assess impact of decomposition on further development of the case e.g. on evidence generation • Assess impact on communication of the case	Clear identification of stakeholder responsibility as decomposed by type of evidence.

The figure below (Figure 9) illustrates as an example how decomposition could be presented in graphical and tabular format based on the table above. More specifically, requirements for sub system A are presented in a table, while the decomposition by sub-system one level above is shown graphically.



or sub - system A have been met - ASCE Node Editor

Insert Format Table

Requirements for subsystem A have been met

Req A.1 has been met	V&V evidence A.1
Req A.2 has been met	V&V evidence A.2
Req A.3 has been met	V&V evidence A.3
Req A.4 has been met	V&V evidence A.4
Req A.5 has been met	V&V evidence A.5
Req A.6 has been met	V&V evidence A.6
Req A.7 has been met	V&V evidence A.7
Req A.8 has been met	V&V evidence A.8
Req A.9 has been met	V&V evidence A.9

Figure 9: Example of balance between graphical and tabular decomposition