White Rose Research Online

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of York.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/165129/

Version: Accepted Version

Proceedings Paper:
Selviandro, Nungki, Hawkins, Richard David and Habli, Ibrahim (2020) A Visual Notation for the Representation of Assurance Cases using SACM. In: International Symposium on Model-Based Safety and Assessment. LNCS. Springer, pp. 3-18.

https://doi.org/10.1007/978-3-030-58920-2_1

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

UNIVERSITY OF LEEDS
University of Sheffield

UNIVERSITY of York

A Visual Notation for the Representation of Assurance Cases using SACM

Nungki Selviandro*[1,2], Richard Hawkins[1], and Ibrahim Habli[1]

[1] Department of Computer Science, University of York, York, YO10 5DD, UK
[2] School of Computing, Telkom University, Bandung, 40257, Indonesia
{ns1162, richard.hawkins, ibrahim.habli}@york.ac.uk

Abstract. The Structured Assurance Case Metamodel (SACM) is a standard specified by the Object Management Group (OMG) that defines a metamodel for representing structured assurance cases. It is developed to support standardisation and interoperability in assurance case development. SACM provides a richer set of features than existing assurance case frameworks. By providing a standardised metamodel for assurance cases, SACM also provides a foundation for model-based assurance case development. For example, model merging can be used to bind packages in complex assurance cases and model validation can be used to check well-formedness of assurance cases. The uptake in the use of SACM has however been slow. The lack of a visual notation for representing SACM arguments has been a major factor in this. As part of the updates for version 2.1 of the SACM standard, we developed a graphical notation that addresses this need. Additionally, there are very few publicly available examples of how SACM may be used in practice, with the SACM standard providing only very limited examples. Moreover, there exists little literature that discusses the potential benefits that using SACM can bring for assurance cases. This paper provides, for the first time, an explanation and worked examples of how to use the SACM notation. The paper also discusses the potential benefits of using SACM for assurance case development and review and the need for empirically evaluating these benefits.

Keywords: Structured Assurance Case Metamodel · Model-Based Assurance Case · Assurance case notations · SACM notation

1 Introduction

An assurance case is commonly used to provide a clear and structured basis for analysing and communicating the assurance arguments and evidence regarding a particular system in a specific environment [11]. An assurance case consists of a related set of auditable elements, such as claims, arguments, and evidence, used to demonstrate that a defined system will satisfy particular properties of interest (e.g. safety/security requirements). In a complex system, structured arguments

* Supported by Indonesia Endowment Fund for Education (LPDP)

will normally be large and complicated, therefore the argument and evidence must be clearly documented to ensure they are communicated in a clear and defensible way between the different system stakeholder (such as developers, reviewers, and regulators).

An assurance case can be represented using a textual (i.e. natural language) and/or graphical approach. The Goal Structuring Notation (GSN) and the Claim-Arguments-Evidence (CAE) are two examples of established assurance case frameworks that can be used to represents an assurance case using graphical representations [7]. These two frameworks have been widely adopted in various domains to represent assurance cases, and examples related to their use can be accessed in the literature, for example, in [10, 8, 13, 3].

The Structured Assurance Case Metamodel (SACM) is a specification that defines a metamodel for representing structured assurance cases. It is issued and published by the Object Management Group (OMG) to improve standardisation and interoperability in assurance case development. SACM is built upon the existing and established assurance case frameworks such as GSN and CAE. However, SACM provides a richer set of features, in terms of expressiveness, than existing assurance case frameworks.

SACM was also developed to better support a model-based approach to assurance case development by supporting high level operations such as model validation, model-to-model transformation and model merging. A number of tools provide some support for model-based assurance case development using SACM. A model-based assurance case tool (the Assurance Case Modelling Environment (ACME)) is currently under development that supports the SACM visual notation. Further discussion of the potential advantages of using SACM for creating assurance cases, including model-based assurance cases, and discussion related to the ACME tool can be found in [15].

To support the adoption of the SACM in assurance case development, we developed a visual notation for representing the SACM using a graphical representation. This was accepted by OMG and incorporated into the latest version of the standard [11]. The visual notation of SACM is developed as an alternative representation to the textual form to represent an assurance case using the SACM specification [14]. There are no publicly available examples of the use of the SACM notation. Other than [15] there is also a lack of literature that discusses the potential benefits of using the SACM in developing assurance cases. In this paper we address these gaps by providing examples based on a concrete case study of the use of the SACM notation for the assurance of machine learning for retinal diagnosis. We also discuss the potential benefits of using the features that SACM provides.

The rest of the paper is organised as follows. In Section 2, we briefly discuss different assurance case representation frameworks. In Section 3, we present the SACM notation and explain its use in representing assurance cases. The potential benefits of using the SACM specification are presented in Section 4. Finally, the conclusion of this paper and discussion of further work are given in Section 5.

2 Representation of assurance cases

A convincing argument that is supported by evidence is the core of any assurance cases; therefore, they need to be clearly documented and represented. A typical approach used in representing an assurance case is through free text using natural languages. However, some problems can occur when text is used as the only form for representing assurance cases. For example, the language used in the text can be unclear, ambiguous and poorly structured. The limitations regarding the use of free text as the only medium for representing safety cases have been discussed further in [10].

In order to overcome the limitations of using free text in representing assurance cases, graphical representation approaches have been introduced. Graphical notations such as CAE and GSN have been developed and, currently, widely adopted for representing assurance cases in various domains. In this section, we briefly introduce GSN and CAE and provide an overview of the SACM specification.

2.1 Goal Structuring Notation

GSN is a graphical notation for explicitly capturing the different elements of an argument such as claims, evidence, contextual information, and the relationships between these elements. It is a well established graphical argumentation notation that is widely adopted within safety-critical industries for the presentation of safety arguments within safety cases.

An assurance case in GSN can be documented using a claim (represented using *GSN Goal*) that is supported by sub-claims and evidence (represented using *GSN Solution*). The relationship between claims and evidence, can be defined using the *SupportedBy* relationship. A *GSN Context* can be used to scope the asserted claim, wherein this case, the relationship between them can be defined using an *InContextOf* relationship. When documenting how claims are said to be supported by sub-claims, sometimes, it can be useful to document the reasoning step involved, in this case, we can use the *GSN Strategy*. An assumed statement made within the argumentation can be documented using the *GSN Assumption*. Justification is also can be added to the argument structure using the *GSN Justification* in order to represent a statement of rationale. In GSN, the claim structure of the argument progresses downwards (top-down approach), from most abstract claim, recorded in the top-level goal, to an assertion about some item of evidence, recorded in the lowest goal in the structure. A GSN structure should be a directed acyclic graph where loops are not allowed.

Further explanation and complete documentation of the GSN elements can be found in [5]. Literature that discuss the application of the GSN, for example, can be found in [1, 6, 9].

2.2 Claim-Argument-Evidence

CAE is a graphical notation for representing assurance cases by documenting a set of claims supported by the argument and the related evidence. A *Claim*

in CAE can be defined as a statement asserted within the argument that can be assessed to be true or false. An *Argument* is defined as a description of the argument approach presented in support of a claim, and an *Evidence* is described as a reference to the evidence being presented in support of the claim or argument. Other than these elements, CAE also provides elements such as *Side-Warrant*, a statement about the reason behind why we can deduce the top-level claim from the sub-claims and under what circumstances the argument is valid, and different types of relationship such as *Supports* (relation between Argument and Claim), *Is a sub-claim of* (relation between Sub-claim and Argument) and *Is evidence for* (relation between Evidence and Sub-claim). Literature that discuss the application of CAE can be found, for example, in [3, 2].

2.3 Structured Assurance Case Metamodel

SACM is a metamodel that defines the specification for representing assurance cases. It was developed to support model-based engineering with existing well-established assurance case frameworks such as GSN and CAE. SACM is composed of the following components:

- the Structured Assurance Case Base that captures the fundamental concepts of SACM;
- the Structured Assurance Case Terminology that defines the mechanism to express terminology and concept used in the assurance cases;
- the Structured Assurance Case Packages that defines the concept of modularity in assurance case development;
- the Argumentation Metamodel that defines a metamodel for representing structured argument; and
- the Artifact Metamodel that specifies the concepts in providing and structuring evidence in assurance cases.

In this paper, we focus on the argumentation part of the SACM since it defines the specification for representing the assurance cases. In the next section, the SACM notation is explained along with examples of its usage.

3 Assurance case representation using SACM argumentation notation

To support the adoption of SACM in assurance case development, we developed a visual notation for graphically representing the SACM argumentation specification. The notation was developed using a systematic process we developed for creating visual notations from metamodels [14]. The process is based on the theories of user-centred visual notation design and takes account of existing notations and the hierarchical structure of the elements in the metamodel to create an effective notation. Although SACM models could be represented visually through the transformation of the models to a different argumentation

notation such as GSN or CAE, these other notations do not provide the richness of SACM, and the transformations can be complicated and incomplete. It is therefore highly desirable to provide the ability to visually represent SACM models directly. The developed SACM notation was accepted by OMG and is incorporated into the latest version of the standard [11].

3.1 Basic elements

Structured arguments in SACM are represented explicitly by the Claims, citation of artifacts or ArtifactReferences (e.g. Evidence and Context for Claims), and the relationships between these elements. The Claim and an asserted relationship to connect between Claims and associate a Claim to its supporting context/evidence are defined under the Assertion class in the SACM. It is possible in SACM to associate a Claim to another Claim using a particular relationship type, and it is also possible in SACM to associate a Claim, for example, using a particular relationship type, to a relationship that is used to connect between elements. In this section, there are several examples provided to illustrate this aspect. A Claim in SACM is visually represented using a rectangle where the claim statement can be written within the rectangle, and a unique element identifier is placed at the top-left corner of the rectangle (as shown in Figure 1). This visual representation is influenced by the GSN. It is being adopted to provide a visual clue to, at least, the assurance case notation user who might be familiar with an existing notation.. A Claim can be supported by more than one Claim, and the relationship between them can be defined using an AssertedInference relationship. A line with a solid arrowhead is used to visualise an AssertedInference with a solid dot placed in the middle of the line that can be used as a connection point (as shown in Figure 1).

In some cases, the relationship that associates more than one Claim may not always be obvious. In such cases, ArgumentReasoning can be used to provide a further description of the reasoning involved. An ArgumentReasoning is visually represented using an annotation symbol (as shown in Figure 1). It can be attached to the AssertedInference relationship that connect the Claims.

A Claim may require a reference to contextual and/or evidential information. In SACM, both contextual and evidential information is defined as an Artifact, and a reference to this information is defined as ArtifactReference. Therefore, to cite contextual and evidential information to support a particular Claim, we can use an ArtifactReference. An ArtifactReference is visually represented using a note/document symbol to provide a clue to its actual meaning (as an artifact) and with an arrow placed on the top right of the symbol to indicate the meaning of a reference (as shown in Figure 1).

Since the ArtifactReference as a reference to evidence and contextual information uses the same visual representation, we can differentiate between these in a diagram by identifying the type of relationship that is used to connect the ArtifactReference to its supporting element, and the position of the ArtifactReference relative to its targeted element. For an ArtifactReference that is used as a reference to a Context, the AssertedContext relationship is used to define

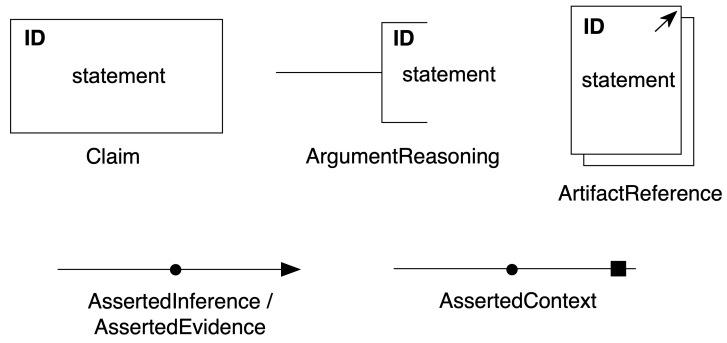

Fig. 1. Visual representation of a Claim, ArgumentReasoning, ArtifactReference, AssertedInference, AssertedContext, and AssertedEvidence relationship

the relationship between the ArtifactReference and its targeted element (e.g. a Claim), and the ArtifactReference is placed horizontally relative to its targeted element. The AssertedContext relationship is visually represented as a line with a solid square placed near to a line-end, and a solid dot placed in the middle of the line that can be used as a connection point (Figure 1). For the ArtifactReference that is used as a reference to Evidence, we can use the AssertedEvidence relationship. The ArtifactReference, in this case, is located vertically (below) relative to its targeted element (e.g. Claim). The visual representation of the AssertedEvidence in SACM is identical with the visual representation of the AssertedInference, and we can differentiate them when used in a diagram by identifying its source element. For the AssertedEvidence relationship, the source element must be an ArtifactReference, and for the AssertedInference, the source element must be a Claim.

To show the use of the above SACM elements in a graphical diagram, in this paper, we adapted an assurance case [12] for a deep learning system used for retinal disease diagnosis and referral [4]. This system comprises 2 different neural networks. The first network, called Segmentation Network, takes as input three-dimensional Optical Coherence Tomography (OCT) scans and creates a detailed device-independent tissue-segmentation map. The second network examines the segmentation map and outputs one of the four referral suggestions in addition to the presence or absence of multiple concomitant retinal pathologies. The adapted assurance case from [12], originally, was constructed and presented using GSN. In this paper, we modify and reconstruct the assurance case diagram and represent it using the SACM notation. Figure 2 shows the use of the above elements in an assurance case fragment diagram.

In Figure 2, the top-level claim is the 'ML Assurance Claim'. It is concerning the performance of the system for providing correct diagnosis and referral decisions. This Claim is supported by two references to contextual information (cited via ArtifactReference 'Clinical Setting' and 'Automated Retinal Diagnosis'). The relationship used to associate the ArtifactReference as a reference to a context

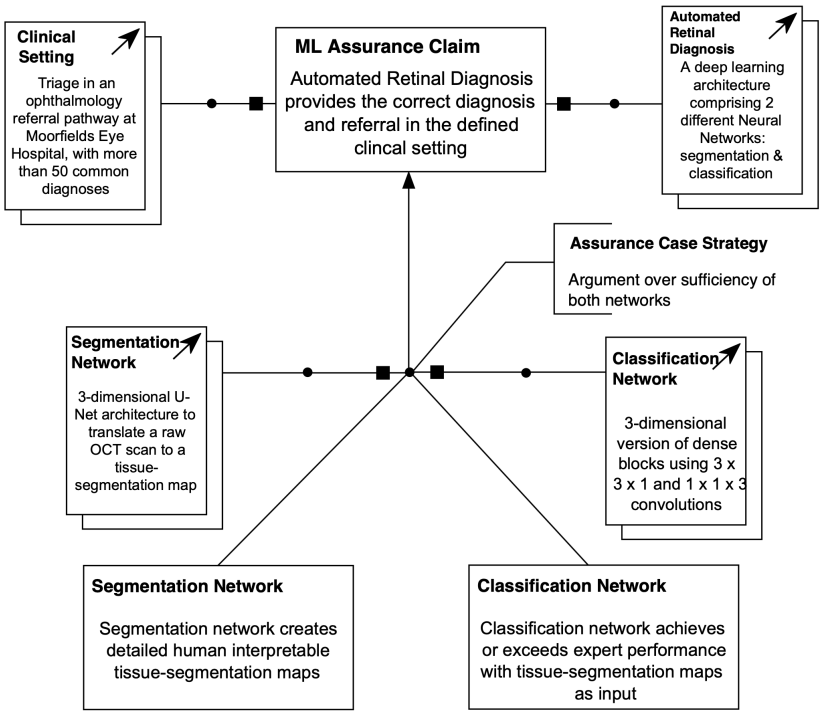

Fig. 2. The used of Claim, ArtifactReference (as a context), ArgumentReasoning, AssertedInference, and AssertedContext in an assurance case fragment

to the supported claim is the AssertedContext relationship. The ArtifactReference, in this case, is placed horizontally relative to the claim since it is used as a reference to contextual information. In order to support the top-level claim, there are two sub-claims added to the structure. The relationship between these Claims is defined using the AssertedInference relationship. There are also two ArtifactReference elements (as a reference to contextual information supporting the relationship between Claims), and also an ArgumentReasoning attached to this relationship to provide further explanation regarding the assertion.

In Figure 3, we show the use of the ArtifactReference as a reference to evidential information. In this case, the 'Classification Performance Evidence' is presented as an ArtifactReference to evidential information. It is located vertically relative to its supporting elements (Claims). The relationship used to associate the ArtifactReference to its supporting Claims is defined using the AssertedEvidence relationship.

In Figure 3, we also can see the use of ArtifactReference as a reference to a context such as the 'Test Data' Artifact Reference that is placed horizontally relative to its supporting Claim. The relationship used to associate this ArtifactReference to its supporting element is the AssertedContext relationship.

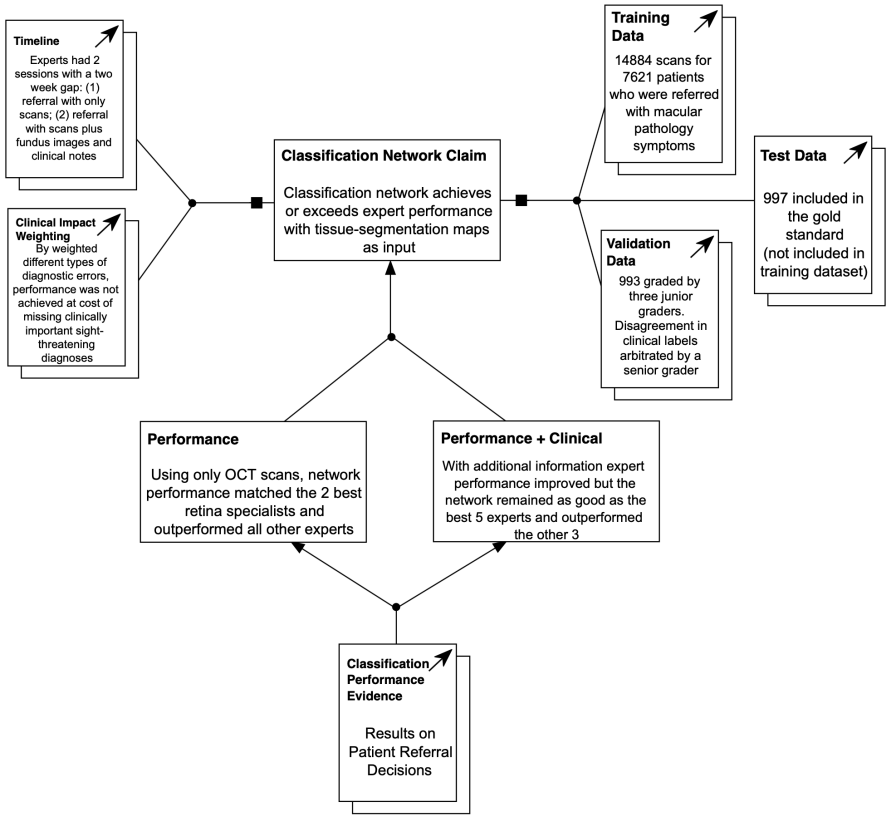

Fig. 3. ArtifactReference as a reference to evidence and ArtifactReference as a reference to context used in a diagram

3.2 Different types of Claim and relationship

An Assertion in SACM can be declared into several types:

- **Asserted** that indicates an assertion is asserted
- **Assumed** indicating that the Assertion being made is declared by the author as being assumed to be true rather than being supported by further argumentation
- **Axiomatic** indicates that the Assertion being made by the author is axiomatically true, so that no further argumentation is needed
- **Defeated** indicating that the Assertion is defeated by counter-evidence and/or argumentation;
- **AsCited** indicating that because the Assertion is cited, the AssertionDeclaration should be transitively derived from the value of the AssertionDeclaration of the cited Assertion
- **NeedsSupport** indicating that further argumentation has yet to be provided to support the Assertion.

The concrete example of an assertion in SACM can be either as a Claim or an asserted relationship. AssertedInference, AssertedContext, and AssertedEvidence are examples of asserted relationships in SACM. Therefore, in this case, these elements can be declared into a specific assertion declaration, as mentioned above. In the context of notation design, each specific assertion declaration, as mentioned above, is designed to fit for each assertion declaration type (i.e. the Claim and the asserted relationship).

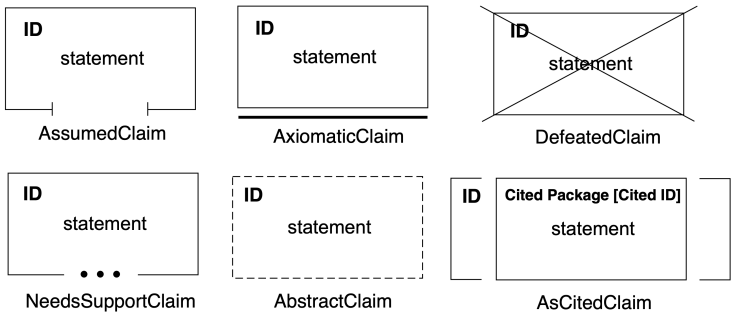

Fig. 4. Different types of Claim in SACM

Figure 4 shows the different types of Claim in SACM. For each different type of Claim, the basic visual representation of a Claim (i.e. a rectangle) is used, a necessary decoration is added to present a particular meaning of the assertion declaration. An AssumedClaim is visualised as a Claim with a line gap on the bottom part of the claim to deliver an assumption meaning which indicates that there is no supporting element that can be attached to this element since it is defined as an assumption. An AxiomaticClaim is visualised as a Claim with a thick line placed below the Claim to indicate that no further argumentation is needed to support this Claim since it is defined as an axiomatic. A DefeatedClaim is visualised as Claim with a cross placed on top of the Claim to indicate this Claim is defeated by a counter argument/evidence. AsCitedClaim is visualised as a Claim placed within square-brackets to indicate that this Claim is a citation claim and further explanation about this claim is presented in another argument structure. The AsCitedClaim notation also can be combined with the others claim, e.g., AsCitedClaim citing an AssumedClaim. A NeedsSupportClaim is visualised as a Claim with three dots placed at the bottom part of the rectangle to indicate further argumentation has yet to be provided to support this claim. A Claim also can be represented as an AbstractClaim. This represents a Claim defined in an abstract form that requires instantiation for a particular argument. Abstract elements, such as claims and relationships, are used to support the construction of assurance case patterns. To provide a visual clue to abstract elements they are rendered using dash-lines, as seen for the AbstractClaim in Figure 4.

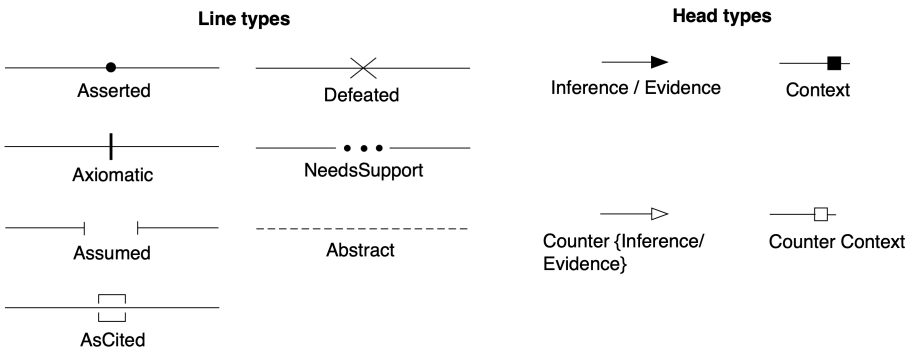

Fig. 5. Different types of asserted relationship in SACM

Figure 5 shows the different types of asserted relationship in SACM. For each type of asserted relationship (e.g. AssertedInference, AssertedContext, and AssertedEvidence), they also can be declared into different types of assertion declaration. Similar visual representation decoration as used in visualising each different type of Claim is used to visualise each type of assertion declaration for the asserted relationship. For example, an Assumed AssertedEvidence is visualised as an arrow headline with a line gap placed in the middle of the line. An Axiomatic AssertedInference is visualised as an arrow headline with a thick line placed in the middle of the line. A Defeated AssertedContext is visualised as a line with a solid square placed near to a line-end and a cross placed in the middle of the line. The use of similar visual representation to represent the same meaning (assertion declaration types) for different elements, such as a Claim and an asserted relationship, was designed to provide a visual clue to the user.

In Figure 5, we also can see a type of relationship that is defined as Counter. This type of relationship can be used to associate a particular element (e.g. ArtifactReference as a reference to evidence) to counter a particular element such as a Claim. The visual representation of a Counter relationship is visualised as a hollow arrowhead line for an AssertedInference or AssertedEvidence, and a hollow square for an AssertedContext relationship. In order to use this relationship in a diagram, we can declare its specific purpose by combining the type of line and the type of the line-head. For example, we can define an AssertedInference relationship by using the Asserted line-type and the Inference head-line-type with a purpose to make an association between Claims. In case we want to define, for example, an Asserted Counter-Evidence relationship, we can use the Asserted line-type and the CounterEvidence line-head type.

In Figure 6, we show the use of AssumedClaim to provide an assumption in supporting a particular Claim. The Claim 'Gold Standard' in Figure 6 is declared without any supporting evidence or argumentation; therefore, this Claim is defined as an AssumedClaim.

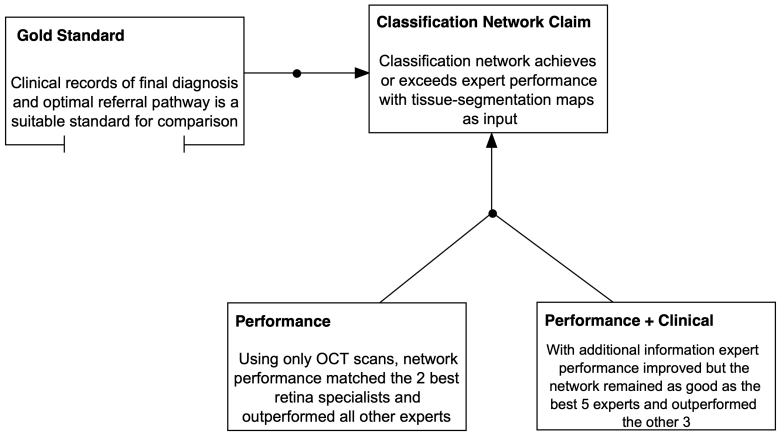

Fig. 6. Example of an AssumedClaim

In Figure 7, we show an example of the use of an AsCited Claim. In this example, the 'Segmentation Outcome Interpretability' Claim is defined as an AsCitedClaim that means it is citing another Claim ('Segmentation Map Result' Claim) that is defined and developed in a different argument structure (e,g, defined in an ArgumentPackage named Arg.Pkg.1) that deals with human interpretability of segmentation maps.

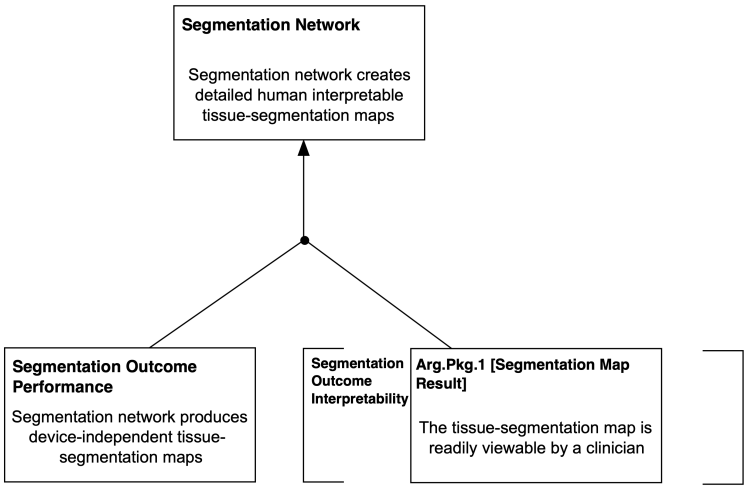

Fig. 7. Example of an AsCitedClaim

In Figure 8, we show an example of the use of the NeedsSupportClaim. In this case, the 'Ambiguous Regions' Claim, that is supported by the 'Segmentation Outcome Performance' Claim, is defined as a NeedsSupportClaim. This means that it needs further argumentation/evidence to be provided.

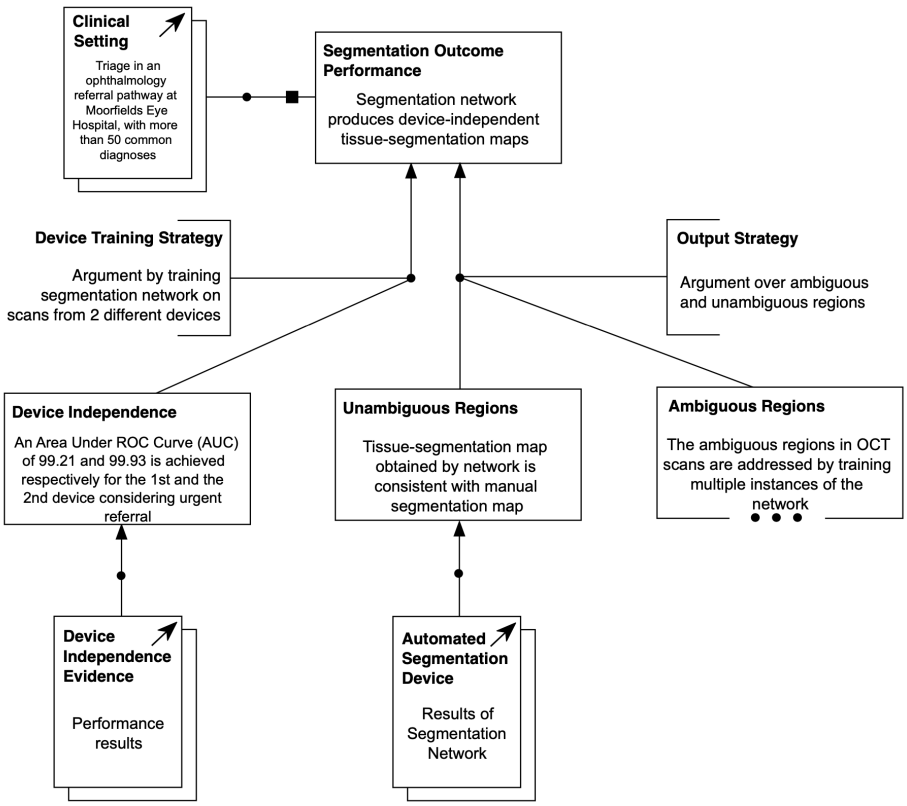

Fig. 8. Example of a NeedsSupportClaim

Figure 9 shows an example of the abstract elements. Claim 'Network' is defined as an AbstractClaim; it concerns the sufficiency of the network in the context of a deep learning system used for retinal disease diagnosis and referral (top-level argument). Claim 'Network' can be instantiated when being implemented in a concrete case such as defining the type of the network (e.g. segmentation network). Similarly, the AbstractArtifactReference that support the Claim can be instantiated further since, in this example, they are defined as an abstract element.

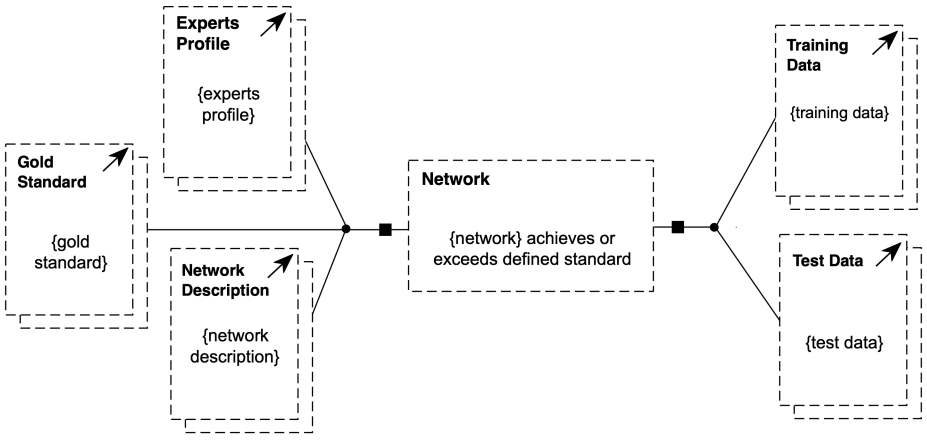

Fig. 9. Example of abstract elements in SACM notation

4 SACM: potential benefits

In the previous section, we have discussed the SACM notation based on concrete examples from a clinical diagnosis assurance case. In this section, we discuss the potential benefits of using this notation in assurance case development and review.

SACM is developed to support standardisation and interoperability in assurance case development. SACM provides a richer set of features than existing assurance case frameworks. There are several potential benefits of using SACM in representing assurance cases relative to the existing assurance case notations, for example:

- representing dialectical assurance case argument through the use of CounterInference/Evidence.
- presenting an argument concerning a particular Assertion in SACM assurance argument via MetaClaim relationship.
- associating a number of argument elements into a common group with a particular interest, such as different views of stakeholder (via ArgumentGroup).

Figure 10 shows an example dialectical assurance argument in SACM. Here, for purpose of illustration, we modify and reconstruct the assurance argument fragment in Figure 8. The 'Unambiguous Regions' Claim in Figure 10 is declared as a DefeatedClaim because it is countered by a counter-evidence cited via ArtifactReference 'Counter-Evidence'. As a result, the tissue segmentation map can not be said to be consistent with the manually-generated one. To associate the counter-evidence (cited visa ArtifactReference) and the targeted claim, in this example, the CounterEvidence relationship is used.

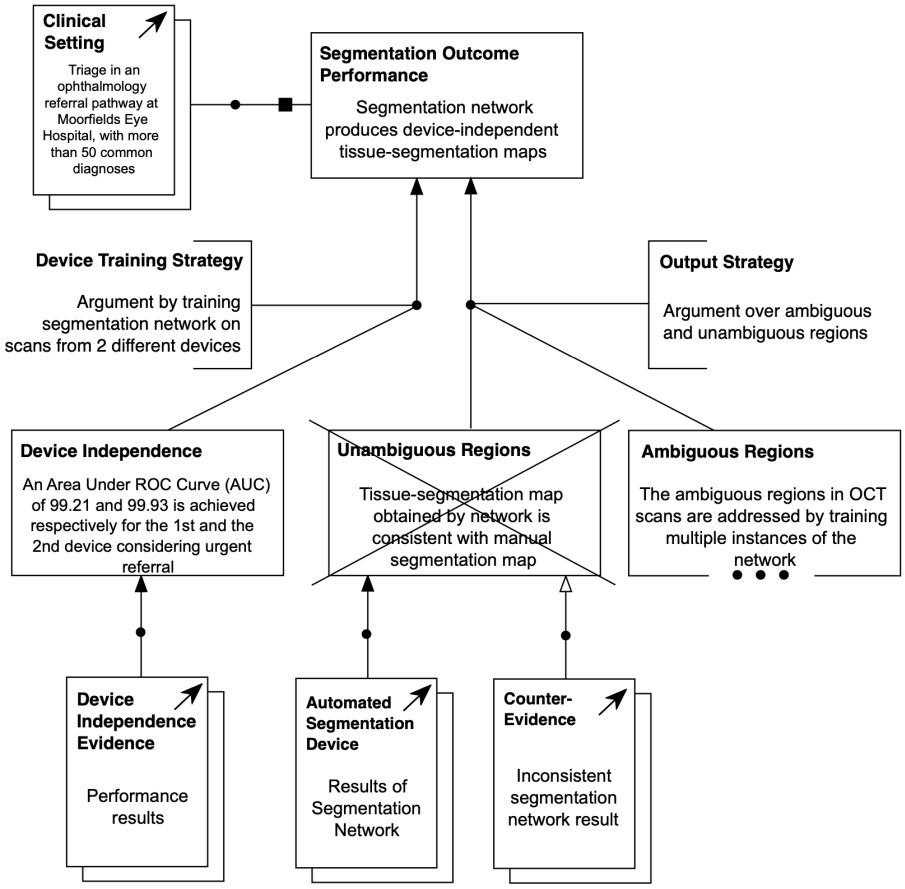

Fig. 10. Example of dialectical assurance argument

5 Conclusion and future work

SACM is a metamodel that defines the specification for representing assurance cases. It was developed to support the model-based assurance case development with existing well-established assurance case frameworks such as GSN and CAE. The adoption of the SACM in presenting assurance cases is considered slow. The lack of a visual notation for representing the SACM arguments has been a major factor in this. We developed a visual notation to support the adoption of the SACM. The notation is developed based on visual notation theories and considered the hierarchical structure of the elements in the metamodel. The developed notation was accepted by OMG and was incorporated into the latest version of the standard.

This paper provides, for the first time, an explanation of how to use the SACM notation in presenting assurance cases. The use of SACM elements such as

Claim, ArtifactReference, and different types of relationship that can be used to associate these elements have been demonstrated and described through several concrete examples from a clinical diagnosis assurance case.

As for the future work, we are currently empirically evaluating the effectiveness of the new graphical notation in different domains. We are also developing model-based tools that integrate the notation into established model-based assurance case frameworks that traceably link the assurance case model with external artefacts, e.g. design models and service data.

References

1. Bate, I., Hawkins, R., McDermid, J.: A contract-based approach to designing safe systems. In: Proceedings of the 8th Australian workshop on Safety critical systems and software-Volume 33. pp. 25–36. Citeseer (2003)
2. Bishop, P., Bloomfield, R., Penny, J., Eaton, A.: A methodology for safety case development. In: Safety-Critical Systems Symposium (1998)
3. Bloomfield, R., Bishop, P.: Safety and assurance cases: Past, present and possible future–an adelard perspective. In: Making Systems Safer, pp. 51–67. Springer (2010)
4. De Fauw, J., Ledsam, J.R., Romera-Paredes, B., et al.: Clinically applicable deep learning for diagnosis and referral in retinal disease. Nature medicine **24**(9), 1342–1350 (2018)
5. GSN: Goal Structuring Notation (GSN) Community Standard Version 2 (2018), https://scsc.uk/r141B:1?t=1
6. Hawkins, R., Clegg, K., Alexander, R., Kelly, T.: Using a software safety argument pattern catalogue: Two case studies. In: International Conference on Computer Safety, Reliability, and Security. pp. 185–198. Springer (2011)
7. Hawkins, R., Habli, I., Kolovos, D., Paige, R., Kelly, T.: Weaving an assurance case from design: a model-based approach. In: High Assurance Systems Engineering (HASE), 2015 IEEE 16th International Symposium on. pp. 110–117. IEEE (2015)
8. Hawkins, R., Kelly, T., Knight, J., Graydon, P.: A new approach to creating clear safety arguments. In: Advances in systems safety, pp. 3–23. Springer (2011)
9. Kelly, T., McDermid, J.: Safety case patterns-reusing successful arguments (1998)
10. Kelly, T., Weaver, R.: The goal structuring notation–a safety argument notation. In: Proceedings of the dependable systems and networks 2004 workshop on assurance cases. p. 6. Citeseer (2004)
11. (OMG), O.M.G.: Structured assurance case metamodel (SACM) Version 2.1 (2020), https://www.omg.org/spec/SACM/About-SACM/
12. Picardi, C., Habli, I.: Perspectives on assurance case development for retinal disease diagnosis using deep learning. In: Conference on Artificial Intelligence in Medicine in Europe. pp. 365–370. Springer (2019)
13. Picardi, C., Hawkins, R., Paterson, C., Habli, I.: A pattern for arguing the assurance of machine learning in medical diagnosis systems. In: International Conference on Computer Safety, Reliability, and Security. pp. 165–179. Springer (2019)
14. Selviandro, N., Kelly, T., Hawkins, R.D.: The visual inheritance structure to support the design of visual notations. In: Third International Workshop on Human Factors in Modeling (HuFaMo'18). York (2018)
15. Wei, R., Kelly, T.P., Dai, X., Zhao, S., Hawkins, R.: Model based system assurance using the structured assurance case metamodel. Journal of Systems and Software **154**, 211–233 (2019)