==

Reviewer #1:
This paper presents how to use SACM (Structured Assurance Case Metamodel), a meta-model being developed in OMG, and briefly shows a tool based on Eclipse which translates from existing graphical notation such as GSN into SACM.

-Thank you very much for taking the time to provide such a detailed review, with these constructive suggestions. We have responded to all the comments with 1) changes to the paper and 2) clarifications. We hope that the revised paper matches to your expectation and is sufficient for publication.

Unfortunately, this paper is not enough for a journal paper:

1. There are many sentences which are not supported by evidence or other arguments. For example,

- In abstract, SACM provides a solid foundation for model-based system assurance, which bears great application potentials in growing technology domains such as Cyber-Physical Systems and Internet of Things.
  -> why it is "a solid foundation"," great application potential", etc,
- Page 7,
  SACM provides a complete solution for model based system assurance case construction. -> I do not understand the meaning of "complete solution". If it is to be "complete solution", then a definition of "complete solution" is needed.

-Thank you very much for your comment.
To further explain our claim in detail, we have added some contents in Section 2.3 summarising the features in SACM that are not supported by GSN ane CAE.

-SACM is developed based on 20 years of experiences in system assurance with two well established argument notations: GSN and CAE.
Being based on these two notations (both with an established track record, including usage for open and adaptive systems) and with additional features (that have been proposed and evaluated by GSN/CAE practitioners), SACM provides a defensible foundation for system assurance for emerging system concepts such as CPS (which is essentially a type of open adaptive system), where model-based assurance cases are the key to system assurance at runtime.
Through the provision of an OMG standard, with the associated obligations of the provision of a fully specified metamodel, SACM provides a clearer basis than GSN and CAE for a model based approach.

-We have changed the section in page 7, with additional explanation on assuring open adaptive system at runtime.

2. It is unclear what are the author's contributions.
The first part of the paper is an introduction of SACM and GSN, which are not new. I think the authors university has been participating in developing SACM in OMG, but the Standard has been already published by OMG.

-Whilst there are many contributors of the standard, the authors are the principal contributors of SACM.
The standard follows the expected form and functions of an OMG standard, but that can be insufficient for people to grasp the meaning the usage of the notations in the standard.
This paper addresses the insufficiency by providing an exposition of SACM via examples.

-The SACM version discussed in this paper is of version 2.0, which was released in May 2018. SACM v2.0 makes rather a large number of enhancements to the standard compared to SACM v1.0, such as: more detailed modularity support, multiple language support, counter arguments, evidence-artefact traceability, describing level of trust for arguments, etc.

-As the originators of GSN, we are aware of how much effort and industry involvement it has taken to reach the level of adoption of GSN over a 25 year period. And the number of organisations have made a significant commitment in making GSN as their primary safety case development technique (e.g. EuroControl safety case development manual).

-We acknowledge that for assurance practice to evolve, model-based system assurance is fundamental. SACM provides a more comprehensive platform towards this direction. Hence we felt the need to explain SACM in detail in order for it to be adopted by industry

The second part of the paper is an example of SACM application and a tool explanation which translates GSN into SACM.

- The example of SACM usage seems new, but it is just briefly described and there is no evaluations how the model with assurance cases is effective.

-We have included a table containing the features in SACM that are not supported by GSN/CAE, with corresponding references to works that motivated such features.

- There are several existing tool such as ASCE(https://www.adelard.com/asce/choosing-asce/standardisation.html) and D-Case Editor(https://github.com/d-case/d-case_editor) which support conversion between SACM and other meta-models.

-We have included a discussion in Section 2.2, which lists a number of model-based assurance case tools, including ASCE and D-Case. We also explained that the support for SACM claimed by these tools are for SACM v1.0, which is replaced by SACM 2.0. Hence, the support for SACM is out-dated.

- Comparison with existing tools are not appropriate. For example, in page 4, "A number of drawing tools have been developed [10-14] to produce GSN diagrams. Although GSN diagrams produced by the majority of the tools are valuable in communicating safety argumentations amongst stakeholders, these diagrams cannot be consumed and interpreted by computers (e.g. automated validation of safety argumentation, automated generation of safety case reports)."
  I think , at least, ASCE tool supports semi-automatic validating function. If the authors really want to say "these diagrams cannot be consumed and interpreted by computers (e.g. automated validation of safety argumentation, automated generation of safety case reports).", more detailed and technical

explanation is needed.

-We have restructured Section 2 to discuss our motivations incrementally. We pointed out the shortcomings of existing tools. For ASCE, the validation functions are embedded in the tool, the model produced by ASCE is not visible to the users, therefore, user defined operations are not permitted.

- Also,
  " Some GSN tools support exporting GSN diagrams into machine consumable models [13,14]. However, these tools implement their own versions of the GSN metamodels, which do not consider the links from safety argumentations to their supporting evidence. Therefore, there is little value in performing model management operations on them. " -> This seems wrong. In Astah GSN support page (http://astah.net/editions/gsn), it says it supports SACM, and if SACM is able to do so as the authors describe, then Astah GSN also can do so.

-Like previously mentioned, the claimed support for SACM for existing tools are out-dated. This transitively out-dates the GSN2SACM transformation provided by these tools. In addition, there is no visibility for the GSN to SACM transformation. If the future is model based assurance case, the transformation from GSN/CAE to SACM need to be visible in the MDE domain (and need to be standardised, too). Currently , the transformations are hard-coded to the tools that support SACM.

Overall, the paper lacks sufficient claims, evidence and arguments. Unfortunately at this time, the paper is not appropriate to be accepted.

-We have added contents in Section 2 to incrementally motivate our work, with references to existing works that back our claims. We provided a summary of motivations at the end of Section 2. We hope these changes are sufficient to explain our motivations for this work. Thank you very much for your time.

====================Turn page for more contents====================

===
Reviewer #2:

The paper revolves around the Structured Assurance Case Metamodel (SACM), which has been put together by the Object Management Group (OMG) to standardize assurance case development. SACM is a more powerful approach than current assurance case models (e.g., GSN and CAE), however, it is still not widely used as its intended usage has not been sufficiently explained, and there has not yet been ways to translate from the more widely used notations, into SACM. The paper therefore aims to explain the intended usage of SACM by giving a detailed exposition of it, and discussing how to use SACM to create assurance case models. The paper also provided SACM compliant metamodels for GSN and CAE, as well as mappings from each of these to SACM via model transformations. Finally, tool support is briefly discussed as part of the Assurance Case Metamodeling Environment (ACME).

-Thank you very much for taking the time to provide such a detailed review, with these constructive suggestions. We have responded to all the comments with 1) changes to the paper and 2) clarifications. We hope that the revised paper matches to your expectation and is sufficient for publication.

Strengths:

There is originality in this work, as no other related work has addressed SACM in this manner, demonstrated its use and detailing transformations from known assurance case notations into SACM.

SACM will soon become the standard assurance case modelling notation, and this work is highly relevant to the assurance case and model based engineering communities, who need to understand the state of assurance case development, and techniques to transform current assurance case notations into SACM.

The paper is overall well-structured and the message of it is clear. The needed background on safety cases, model driven engineering, SACM, GSN, and related work is well presented. An example (a case study on the European Train Control Systems) is presented to explain the approach, and tool support is also briefly demonstrated through the ACME environment.

Weaknesses:

The authors present in Section 4 the packages defined in SACM. It was not clear whether these already exist as part of OMG's initiative, or if they are contributions made by the authors themselves.

-Thank you very much for your comments. The SACM components described are of SACM version 2.0, which was released in May, 2018. We are the principal contributors of SACM v2.0. The standard follows the expected form and functions of an OMG standard, but that can be insufficient for people to grasp the meaning of the notations in the standard. Therefore, we try to address this insufficiency by providing an exposition of SACM via examples.

The authors present an approach for "assurance case composition" for incremental safety (the example in Section 5.4) via package "binding". However, it is well-known that safety is not a compositional property, and emergent behaviour can arise and needs to be taken into consideration. One has to be careful making claims about composing assurance cases without paying attention to these issues, which remain as challenges in the safety/assurance community. A clear statement should be made about these issues and how the authors plan to address in the future.

-The example is kept simple for illustrative purposes. In reality, it is important to be careful that in modular safety case construction, we do not simply assume that the safety of the whole system is guaranteed by a composition of arguments about the safety of the parts. However, modular safety case construction makes no such automatic assumption. It remains the responsibility of the safety case developers to ensure that each safety case module correctly identifies its dependency on other modules that must be satisfied in order to assure the safety of the composed system.

-However, the aim of this work is to illustrate how the facilities in the SACM can be used, we therefore added an description at the end of Section 5.4 to clarify the premise of safety case composition.

Other issues that this paper does not address, which I was hoping to see:
1. The need for strategies to connect leaf goals to solution nodes in GSN, and how that is enforced in SACM.

-The role of Strategies in GSN has always been explanatory of the inferential reasoning that connects parent Goals to child Goals (In fact, connecting a Strategy to Solution with SupportedBy is forbidden in the GSN standard).This role has been clarified in SACM by making it clear that ArgumentReasoning (a description of the AsertedInference that connects a parent Claim to child Claims), can be optionally connected to any AssertedInference. In this way, ArgumentReasoning in SACM performs the role of Strategy in GSN.

2. As systems change, they require recertification or incremental certification. How does SACM provide support for that, including support for change impact assessment on the assurance cases? An example addressing this would have been really interesting to see.

-SACM, in line with established practice for GSN, supports maintenance of safety arguments through fine grained traceable links between argumentation elements (e.g. Claims) and artefacts.
-As introduced by Kelly in 2003 [1], one of the most effective means of supporting/enabling incremental certification is the ability to support well defined modular safety case interfaces and contracts. As discussed in Section 4, this is supported in SACM.
-We felt that incremental certification in SACM would be out of the scope for this work. We would carry out research on incremental certification with SACM for open adaptive systems at runtime.

Other detailed comments:

1. Several spelling/grammar mistakes, for example:

- in abstract "to promote model based approach" -> "to promote a model based approach"

Note: this occurs in numerous other places in the document, please check.

- keywords "Mode Driven Engineering" -> "Model Driven Engineering" - Fixed
- Page 2 "anc" -> "and" - Fixed
- Page 2 "attracts" -> "has attracted" - Fixed
- Page 3 "with clear" -> "with a clear" - Fixed
- Page 4 ", Model-to-Text Transformation" -> " and Model-to-Text Transformation" - - Fixed
- Page 5 "There is also a need to interoperate from GSN to SACM" -> perhaps "translate" is a better way to describe this rather than "interoperate". - Fixed
- Page 5 "elementswhich" - > "elements which" - Fixed
- Page 5 "it is a good practice" -> "it is good practice" - Fixed
- Page 5 "Due the reasons" -> "Due to the reasons" - Fixed
- Page 5 "to reason the dependability" -> "to reason about the dependability" - Fixed
- Page 7 "consists two components" -> "consists of two components" - Fixed
- Page 7 "When integrating system safety case" -> "When integrating a system safety case" - Fixed
- Page 7 "to re-use good practice of safety cases" ->needs rewording - Fixed
- Page 7 "the use of patterns are discussed" -> "the use of patterns is discussed" - Fixed
- Page 8 "which enables the users" -> "which enable the users" - Fixed
- Page 8 "an Goal" -> "a Goal" - Fixed
- Page 9 Fig 6, S1 has a typo "implmented" and also I think should mention {system X} - Fixed
- Page 9 "example of GSN pattern" -> example of a GSN pattern" - Fixed
- Page 9 "is often an manual process" -> " is often a manual process" - Fixed
- Page 10 remove "In overview" in Section 4.1 - Fixed
- Page 13 'hazard' (please fix quotations) - Fixed
- Page 15 'URI' (please fix quotations) - Fixed
- Page 17 type in yellow box in Fig 14 (in general, please check all figure contents for typos) - Fixed
- Page 18 "create an Category" -> "create a Category" - Fixed
- Page 19 'cite's -> cites - Fixed
- Page 19 "element that extend" -> "element that extends" - Fixed
- Page 20 "An detailed" -> "A detailed" - Fixed
- Page 21 "The equivalence of this Claim" -> "The equivalent of this Claim" - Fixed
- Page 24 "figure 27" -> "Fig.27" (please make sure all references to figs are fixed) - Fixed
- Page 25 "ArtifactRefernce" -> "ArtifactReference" - Fixed
- Page 25 "For example, assurance case developer" -> "For example, an assurance case developer" - Fixed
- Page 25, "Sometimes, a justification of AssertedRelationships can be attached to further
clarify the reasons of the AssertedRelationship, ArgumentReasoning is used for

this purpose." -> ""Sometimes, a justification of AssertedRelationships can be attached to further
clarify the reasons of the AssertedRelationship, then ArgumentReasoning is used for this purpose." - Fixed

- Page 26 "that enable re-use of good practice in safety cases" - needs rewording. - Fixed
- Page 26 "instantiated, " -> instantiated, and" - Fixed
- Page 28 ", the ArgumentReasoning S1" -> ", and the ArgumentReasoning S1" - Fixed
- Page 28 "ImplmentationConstraint" - "ImplementationConstraint" - Fixed
- Page 28 "and model management engine" -> "and a model management engine" - Fixed
- Page 30 "that binds" -> "that bind" - Fixed
- Page 30 "comparing to" -> "compared to" - Fixed
- Page 31 "so that not only the GSN metamodel can…" -> "so that not only can the the GSN metamodel…" - Fixed
- Page 31 "that binds" -> "that bind" - Fixed
- Page 32 "the the" -> "the" - Fixed
- Page 32 "Most part of the…" -> "Most of the…" - Fixed
- Page 33 "This requires that analysis to be performed…" -> "This requires analysis to be performed…" - Fixed
- Page 34 "…in CAE provides…" -> "… in CAE provide…" - Fixed

2. Page 2 "A number of tools exist…" - please provide references. - Fixed
3. I found the paper mixing between "goals" and "claims". E.g., Page 7 "The purpose of any goal structure is to show how Goals are successively broken down into sub-Goals until a point is reached where claims can be supported…" Please make this consistent throughout paper. - Fixed
4. Example in Fig. 3 is missing a strategy to decompose G1 into G2 and G3, as well as strategies to connect leaf goals to solutions. I understand that the latter is not needed in GSN, but perhaps the former needs to be addressed?
- Strategies are complementary elements which provides the reasoning of decomposition (translated to ArgumentReasoning in SACM), therefore are needed only when description is needed (this example is taken from the GSN standard).
5. Page 10 - please explain what you mean by "consumable by computers"
- we have changed this sentence, to explain that most artefact produced manually are not model-based
6. 4.1 SACM Overview- why is the order the sections are presented not the same order they appear in?
- This has been rectified
7. Figure 12 (consider elements filled in white first) - what does this mean?
- This is a mistake, this sentence is removed now
8. Can you have fine-grained traceability (i.e., to parts of document artifacts that serve as evidence, rather than to entire documents)?
- Yes, this can be achieved in two ways. One way is to specify a model element like EMF does, using model file path + intrinsic ide (the position of the model element, and its parent, if any), another way is to use ImplementationConstraint to specify a model query using languages such as OCL. We are cureently investigating the feasibility of both, therefore it is not written in the paper.

9. Page 17, Fig 14, box on the left refers to "have been elininated" -> should be "eliminated", but wondering why this phrase appears in the first place at all as it is not in the original claim.
- This is accurate, it is rectified now
10. Example used in 4.5 is said to be not really realistic, why not use a better example?
- This is a straight forward example which is easier to undrestand, a more concrete example follows
11. Section 4.6, there is the use of "ArgumentPackage" and "ArgumentationPackage" - are they the same thing? Please make consistent, and if they are not, perhaps differentiate better.
-They are different terms, to differentiate them better. We have used “AssuranceCase Component”, “Base Component”, “Artifact Component”, “Terminology Component” and “Argumentation Component” to denote the components provided by SACM.
12. It is worth mentioning that defeated claims and asCited claims (Figs 25 and 26) have no equivalents in GSN, which is why they don't appear in the figs.
- Very useful suggestion, sentences have been added to clarify this
13. Footnote on page 24 mentions that the descriptions details are omitted due to space limit. Do you mean limit of the figure? This is a journal paper, I am not sure there is a page limit.
- The object model is too complicated if the whole of it is shown, we have removed “due to space limit”
14. Example provided in 5.4 was not detailed enough, in particular, since this presents an assurance case composition/integration scenario, I would have liked to understand the semantics of the integration and just how the package binding works. How safe is the result of the binding? How much confidence can you have about it? What about emergent behaviour (not shown in this example), how can it be detected and handled properly?
-The semantics of binding are the same as argument decomposition. What we mean by that is saying that, e.g. to argue that one claim supported by a foreign
In this sense, for example, there is no type distinction between binding two modules A and B, with a binding C, and re-writing that composition as a monolithic argument structure that connects the arguments in A and B with a reasoning (Argument) presented in C. Like previously discussed, safety composition is a complex task, which involves handling subtle dependencies among systems, we only illustrate how the binding mechanism works in this work.
15. Reference to analysis in Algorithm 1 on page 33 is not clear - where in the algorithm is the analysis? Also, there is another reference to model analysis at the end of page 33, what kind of analysis and why?
- This is an accurate observation. Algorithm 1 actually is the transformation rule Strategy2ArgumentReasoning. We have added descriptions to the algorithm to make it clear how to transform a Strategy to an ArgumentReasoning.
16. In section 7.1, Figure 38 demonstrated how Artifacts can be created/edit - what types of artifacts are supported?
- The types of artifacts are the ones defined in the Artifact package of SACM, we have added explanatory contents at the end of this sentence to remind the reader of what artifacts can be created

17. Page 37 (in Future Work) last paragraph, you mention investigating automated model validation to validate GSN/SACM models. What does it mean to validate them? And how will you do it?
- By automated model validation we mean to check the well-formedness of the model. For example, Strategy-Solution is forbidden, the validation should check against such mistakes.

[1]. Kelly, Tim, and Simon Bates Meng. "The Costs, Benefits, and Risks Associated With PatternBased and Modular Safety Case Development" In *Proceedings of the UK MoD Equipment Safety Assurance Symposium*. 2005.

====================Turn page for more contents====================

Reviewer #3:

-Thank you very much for taking the time to provide such a detailed review, with these constructive suggestions. We have responded to all the comments with 1) changes to the paper and 2) clarifications. We hope that the revised paper matches to your expectation and is sufficient for publication.

It should be mentioned the difference of the expressive power between SACM and GSN. Authors just explained the transformation from SACM to GSN. I wonder that there are elements of SACM that are not be expressed in GSN.

-This is very accurate observation. For clarification, we have added a discussion in Section 2.3, to list the features included in SACM but not in GSN/CAE. Together with it, we added corresponding references to works that motivate these features.

Section 4 and 5 contains similar statements and diagrams. Authors should omit the redundant parts and briefly describe the main point

-Thank you very much, we have removed the redundant parts and kept the description of the elements minimal.

Although Section 7 introduced a tool named ACME, there was no evaluation of the tool. It is necessary to discuss the effectiveness of the tool.
It is meaningless to write what a tool can do.

-ACME is currently a work-in-progress. What we want to show in this section is what can be achieved with model-based assurance cases. For example, with model-based assurance cases, interoperability from GSN/CAE to SACM is enabled; automated validation to check the well-formedness of assurance cases, etc.
-ACME is the first step towards an integrated environment for model-based assurance cases with SACM, whilst providing backward-compatibility with existing approaches (GSN/CAE). We only briefly mention the tool, it would be another paper to discuss in detail ACME.

In conclusion, authors mentioned that SACM provides means of system assurance for Cyber-Physical Systems and IoT. However, there is no evidence of the claim in the paper. Especially, it should be clearly discussed on the meta model elements on SACM for runtime system assurance.

-We have changed Section 2.4 to briefly explain why model-based assurance cases are the key to assure Open Adaptive Systems (including Cyber-Physical Systems) at runtime.
We feel that detailed discussion on runtime assurance cases for Open Adaptive Systems would fit into another paper, therefore removed the references to CPS and IoT in the paper.