

Adoption of Opacity-based Graphical Highlights as a Representation of Structured Assurance Case Metamodel ArgumentGroup

M Haikal Abdussalam*, Nungki Selviandro, Gia Septiana Wulandari

Faculty of Informatics, School of Computing, Telkom University, Bandung, Indonesia

Email: ^{1,*}muhamad2haikal99@gmail.com, ²selviandro@telkomuniversity.ac.id, ³giaseptiana@telkomuniversity.ac.id

Email Penulis Korespondensi: muhamad2haikal99@gmail.com

Abstract—An assurance case is a means of facilitating communication, analysis and information exchange between stakeholders involved in the system security and application development process, depending on its purpose and context. There are several frameworks that can be used when creating assurance cases, one of which is the Structured Assurance Case Metamodel Notation (SACMN), which has a more expressive notation than other frameworks. When creating an assurance case, it is necessary to consider the ArgumentGroup, where all elements are categorised as a group. The SACMN ArgumentGroup has been represented visually using dotted lines and there are still shortcomings in the process of understanding assurance cases and their visualisation. Thus, the opacity-based graphical highlighting technique is an alternative in visually representing the ArgumentGroup by manipulating the opacity on the notation of relevant or irrelevant elements. Therefore, this research was conducted with the aim of observing users in terms of speed, effectiveness and accuracy in answering statements with perceived usefulness, perceived ease of use of ArgumentGroup in SACMN applications and development of existing SACMN applications. Based on the research conducted, it can be seen that the results of the questionnaire testing in relation to the user statements of ArgumentGroup in terms of perceived usefulness and perceived ease of use have a value of 0.705 and 0.727, which are included in the reliable category as a valid questionnaire. With a reliable and valid questionnaire, the results obtained from users using opacity highlights have better scores than dotted lines and can be categorised as successful for adoption as a visual representation of ArgumentGroup SACMN with the highest scores of 25 and 20.

Keywords: Assurance Case; Opacity; Structured Assurance Case Metamodel Notation; ArgumentGroup

1. INTRODUCTION

An assurance case is a very important tool in facilitating the system or application development process, which is used to analyze, communicate and exchange information between stakeholders through arguments and evidence in accordance with predetermined requirements. The creation of an assurance case involves several elements that help explain the statement of claims about the system or application that has been made, as well as arguments and evidence that describe the extent to which the system or application meets the requirements that have been set [1].

The assurance case form has two criteria in the modeling language: textual using natural language and visual using predefined graphical notations. Graphical representations in assurance cases are often used because they are easier to understand and more effective in explaining documented information from structured arguments and existing evidence [2].

In assurance cases, there is a framework that is relatively new and still evolving from year to year that can be used in making assurance cases, namely the Structured Assurance Metamodel (SACM), two other frameworks that are standard and popular graphically often used in making assurance cases, namely Goal Structured Notation (GSN) and Claim-Argument-Evidence (CAE) [3]. However, among these three frameworks, SACM has a more varied and visually expressive notation of elements and relationships than the other frameworks [4]. SACM can also be represented using a visual notation commonly referred to as Structured Assurance Case Metamodel Notation (SACMN). SACMN provides several notations of elements, such as claims and arguments, that are used specifically in the creation of assurance cases [5].

In addition, SACMN has a feature for grouping all notations of arguments, claims, and other elements that have been identified as a group, namely ArgumentGroup. SACMN ArgumentGroup is represented visually by using a dotted line to assist in making decisions from each assurance case [6]. However, the use of dotted line ArgumentGroup still has shortcomings in helping stakeholders to understand the context and purpose of assurance cases, especially in the medical field [7], [8]. There is one of the latest techniques that is widely used for grouping an element in the same modeling language as ArgumentGroup is the opacity highlight technique, and has been implemented in a different framework, namely Business Process Modeling Notation (BPMN) [9]. As for the opacity highlight technique in BPMN, it is represented visually using opacity-based graphical highlights by using opacity manipulation on relevant element notations, which are highlighted while irrelevant element notations are ignored [10]. In this case, opacity-based graphical highlighting is adopted as a representation of the SACMN ArgumentGroup to improve the effectiveness of understanding, communication, information exchange among stakeholders and the accuracy of problem-solving in the context of assurance case modeling in a graphical editor Web application that is still in the development stage, namely SACMN [11]. When adopting technology in existing applications, it is necessary to evaluate it using the Technology Acceptance Model, which is influenced by two main factors, namely perceived usefulness and perceived ease of use [12]–[14]. Therefore, this research aims to adopt opacity-based graphical highlights on SACMN ArgumentGroup and evaluate the use of ArgumentGroup outline border with opacity highlights in terms of informatics student users, namely perceived usefulness and perceived ease of use.

The organization of writing in this paper contains several sections, in section 2 we discuss research methodology by explaining the stages of adopting opacity-based graphical highlights to represent structured assurance case metamodel notation ArgumentGroup. To add some context, we also show theory or literature that supports this research. In section 3, we show results and write some discussion about the result. Finally in chapter 4 we conclude the main result of the research and write some suggestions for further research.

2. RESEARCH METHODOLOGY

SACM stands for Structured Assurance Case Metamodel, a modeling language used for assurance cases. SACM belongs to a relatively new category of modeling languages and is still under development. The advantage of SACM lies in the more varied and expressive notations compared to the other two modeling languages, namely GSN and CAE. When used graphically, SACM-based assurance cases are often referred to as SACMN. Although there are similarities with the other two modeling languages in terms of claims and arguments, SACM has more detailed definitions for these components, such as Claim, ArgumentReasoning, ArtifactReference, AssertedInference/AssertedEvidence, and AssertedContext [11]. In SACMN, there are various notations that have defined usage rules to ensure that assurance case modeling can be well-defined and in accordance with the modeling objectives. The following is an explanation of some of the initial notations contained in SACMN.

On the other hand, the Opacity-Driven Graphical Highlight (opacity highlight) approach is a visual representation approach used in the business process model notation framework [10]. Highlight opacity can be used to improve understanding and problem-solving by manipulating the highlight opacity so that irrelevant elements become transparent while relevant ones should be highlighted. To identify relevant elements, opacity highlights are divided into two categories, namely structural highlight opacity (handles a specific element) and behavioral highlight opacity (handles the order of elements in a partial workflow) [9].

Then, to adopt opacity-based graphical highlights to represent structured assurance case metamodel notation ArgumentGroup, we use stages of research, that are: identifying required application features, designing application features, implementing application features, evaluation, testing results and analyzing test results to develop applications clearly and equipped with flowcharts, as shown in Figure 1.

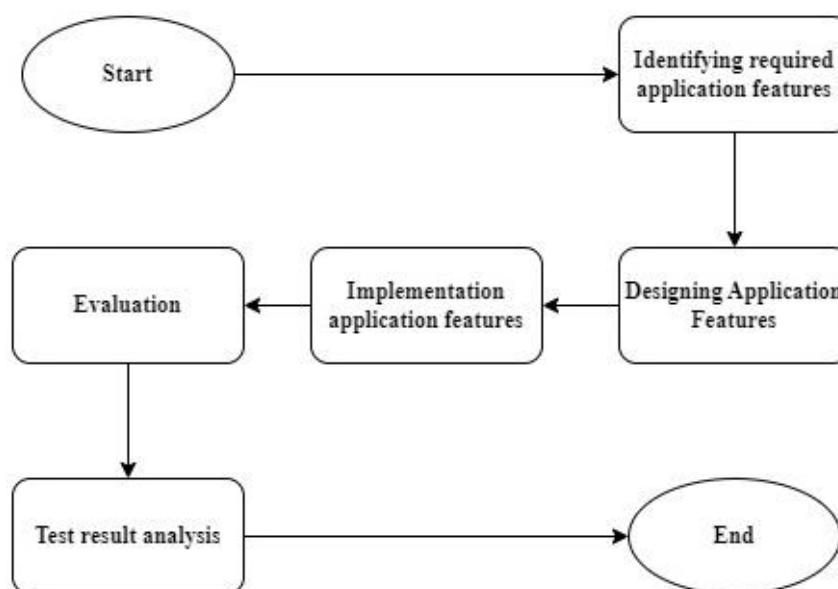


Figure 1. Research Stages

2.1 Identifying required application features

At this stage, the needs of the features to be built are determined from the perspective of feature and user needs. The determination of these needs is completed with three components:

- Feature requirements, which define functional and non-functional feature requirements. From the functional aspect, the system provides features that allow users to select notation elements and relationships that have been defined in the application. Features can be used with several assurance case models, including GSN, CAE, and SACMN for non-functional feature requirements.
- User requirements that satisfy user goals, input and output of users. From the aspect of user goals, namely the success of users using ArgumentGroup to create an assurance case model that wants to be understood and easy to understand, while from the aspect of user input and output will help users to open and save the results of their work. In addition, user-created files can be imported and exported in formats that visualize the results of their work, such as png or svg.

- c. The rules for using ArgumentGroup elements, namely outline borders and opacity highlights from SACMN, are well-defined when modeling assurance cases. This requirement must be met so that assurance case modeling is easy to understand and not confusing.

2.2 Designing Application Features

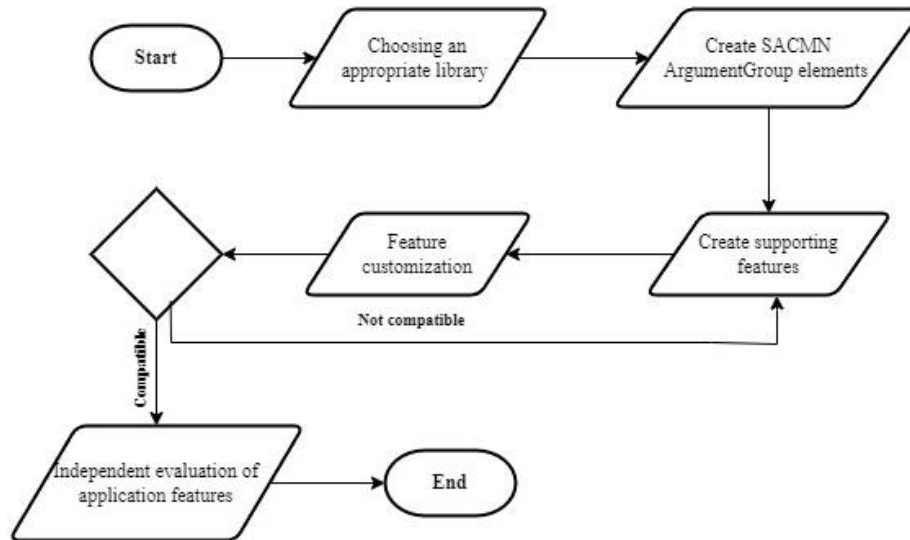


Figure 2. Application Feature Design Stages

In Figure 2, it is shown the stages of designing application features from the development of existing applications until the application features are successfully implemented and tested [11], [15], [16]. The following are the stages of the application development flow.

- Choosing an appropriate library - The library used to develop the charting application is from JointJS using the Rappid extension. JointJS was chosen as the library because it is easy to use from the command line, open source and interactive.
- Create SACMN ArgumentGroup elements - Manually create the ArgumentGroup using the available libraries. The supporting elements of the ArgumentGroup form are first created and adapted to the SACMN notation standard.
- Create Supporting Features - Create the application features required by the application to facilitate the use of the application, such as selecting the required elements, customising the ArgumentGroup form and layout of the SACMN elements, creating outline border and highlight opacity dialogs with different values.
- Feature customization – customization of the application features that have been built and retested to the features that have been created to find out if there are any bugs or errors.
- Independent evaluation of application features - conducting independent testing of built application features by looking at the assurance case model shown in the figure and evaluating them using the user perception aspects of TAM, namely perceived usefulness and perceived ease of use.

2.3 Implementation Application Features

This stage implements the application feature design flow prepared to meet the application system requirements. The following are the device specifications used to develop the application system.

- AMD Ryzen 3 5300U
- RAM 8 GB
- SSD 512 GB
- AMD Radeon™ Graphics

2.4 Application Features Testing

The main purpose of this stages is to determine the level of success and usability of the built application features. Testing application features will refer to a user focused evaluation study using the Technology Acceptance Model method. Testing the features of this application is needed to find out whether the success built is correct or not. Testing this application will require a minimum of 30 Telkom University Informatics study program students to determine the validity and reliability of the success of application development. The following is a scenario for testing the application.

Table 1. Test scenario

No	Testing Scenario Stage
1	Explanation of the assurance case
2	Explanation of the SACMN application and the rules for using ArgumentGroup
3	Users provide a website link to see the appearance of the SACMN application

- 4 Users give an example of the SACMN assurance case model
- 5 The user will be prompted to regenerate the SACMN assurance case model
- 6 Users are asked to group elements of the SACMN assurance case model using an outline border
- 7 Users will be asked to group elements from the SACMN assurance case model using opacity highlights
- 8 Users will be given a questionnaire using the Google Forms platform

The test scenario in Table 1 provides stages for users to test the application. The user is given a short and clear explanation of the assurance case, the SACMN application and the rules for using ArgumentGroup. After the user performs the test, the Technology Acceptance Model (TAM) component is tested, which has aspects of perceived usefulness, which tests the user's perception of the usefulness of the outline border and opacity highlight, and perceived ease of use, which tests the user's perception of ease of use after using the opacity highlight application.

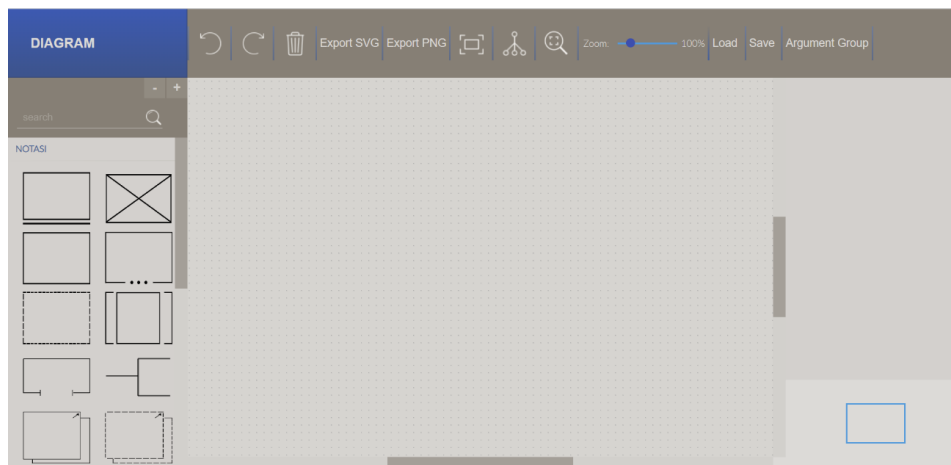


Figure 3. View of the SACMN application

Figure 3 explains that after the user has been given an explanation of the assurance case, the rules for using the SACMN application and a website link to view the application as conveniently as possible. In addition, the user is asked to recreate the assurance case by being given an example of the SACMN assurance case model, which can be seen in detail in Figure 4.

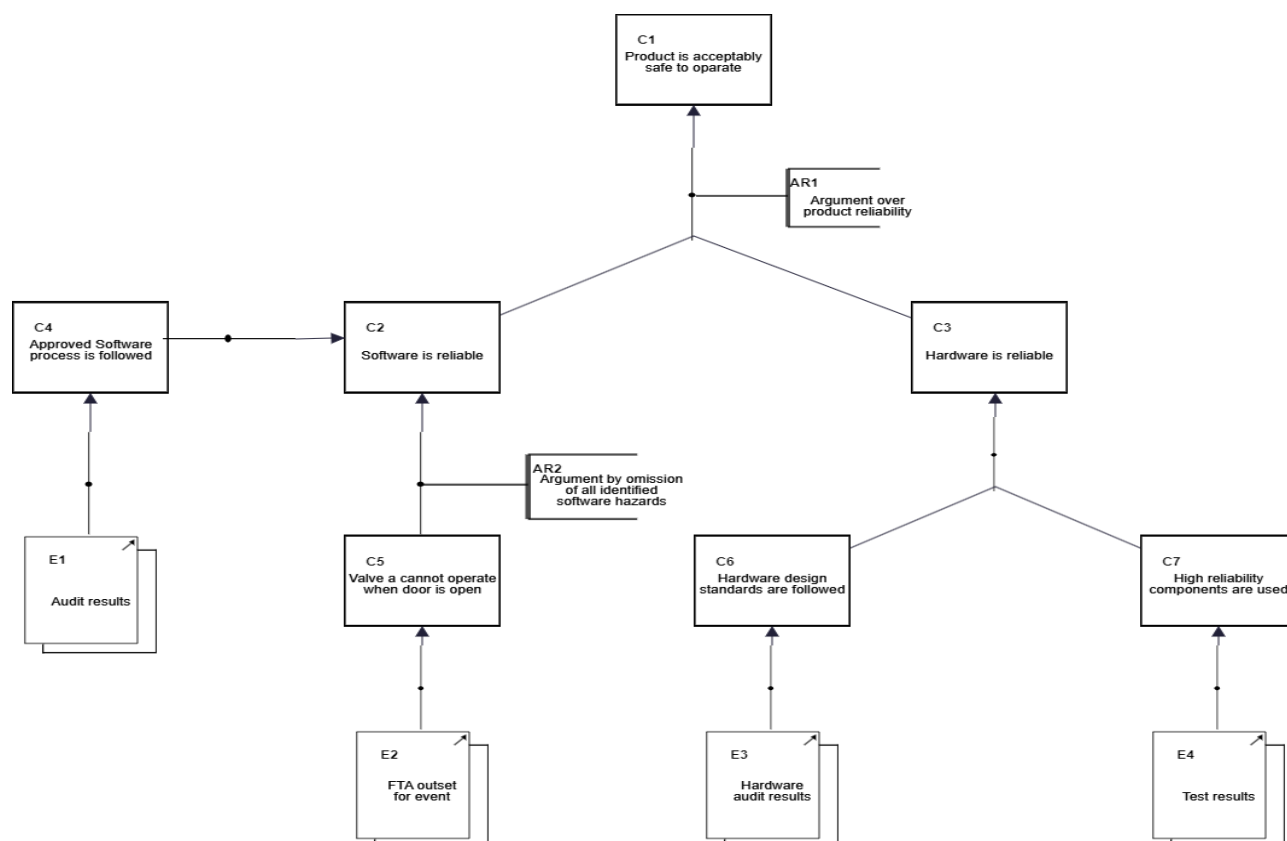


Figure 4. Model assurance case SACMN

In Figure 4, it can be explained that after the user has finished trying to recreate the SACMN assurance case, the user will be asked to group the elements that have been identified. The first part tests the outline border feature, and the second part tests the opacity highlight feature. The following two features will be tested.

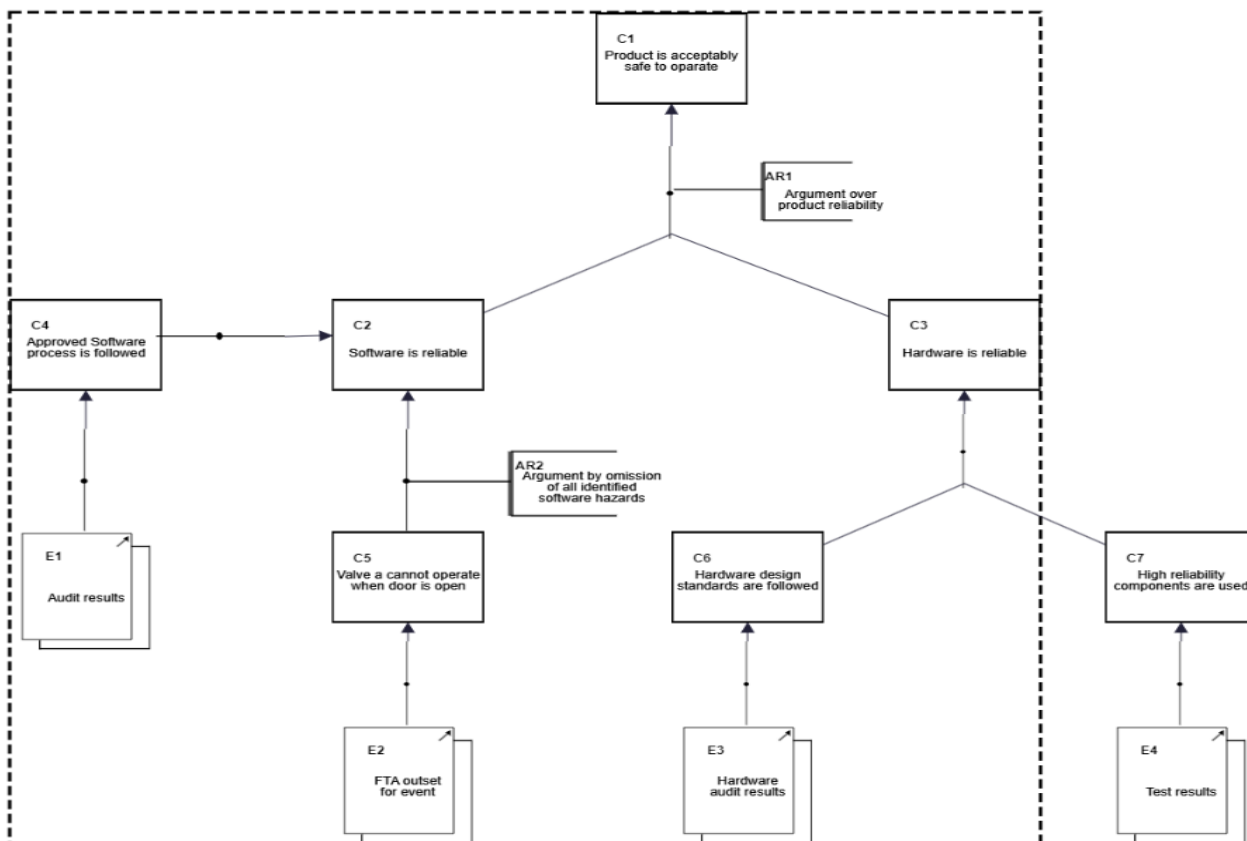


Figure 5. Outline Border

In Figure 5, outline border, after modelling the SACMN assurance case, the user continues to use the outline border feature, which is a built-in ArgumentGroup from the SACMN application, to group important elements using a dotted line.

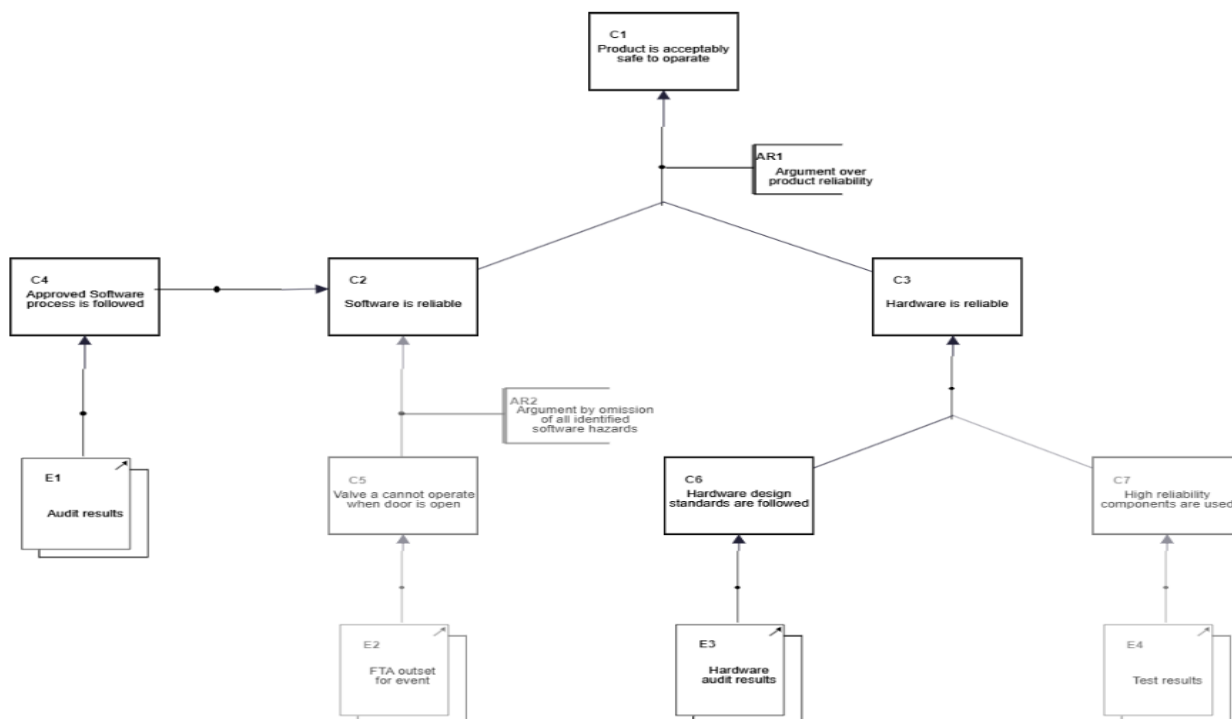


Figure 6. Opacity Highlight

In Figure 6, Opacity Highlight, after the user has modeled the SACMN assurance case, they continue to use the Opacity Highlight feature, which is taken as an ArgumentGroup from the SACMN application, to group elements by manipulating the irrelevant opacity with a value of 0.5 (50%).

After the user has finished modeled the assurance case using the outline border feature and the opacity highlight feature, the user is given a questionnaire. Part of the questionnaire will test perceived usefulness (PU) and perceived ease of use (PEOU) using a Likert scale from 1 for strongly disagree to 5 for strongly agree. This Likert scale will be used for statements related to perceived usefulness and perceived ease of use. The questionnaire is also tested for reliability to ensure that the statements are consistently and stably emphasized to the respondents in each response. Respondents who participated in the questionnaire testing are shown in Table 2.

Table 2. Questionnaire test respondents

Case Processing Summary			
		N	%
Cases	Valid	30	100.0
	Excluded	0	.0
	Total	30	100.0

From the questionnaire testing table, it can be seen that the number of respondents used in the SPSS calculation is 30 respondents from Telkom University computer science students, and there are no missing data, so the sample used remains 100% in accordance with the actual situation. In reliability testing, there are two aspects that will be measured by SPSS, namely perceived usefulness and perceived ease of use.

3. RESULT AND DISCUSSION

In this section, after users have tested the application features, they are asked to fill out a questionnaire via Google Forms. This questionnaire is designed to evaluate perceived usefulness and perceived ease of use. The results of the perceived usefulness (PU) questionnaire, the results of the perceived ease of use (PEOU) questionnaire, and the results of both questionnaires were tested using Cronbach's alpha to determine whether the two questionnaires could be considered reliable and valid. Cronbach's alpha rating categories are shown in Table 3.

Table 3. Cronbach's alpha rating categories

Value	Internal consistency categories
$0.9 \leq \alpha$	Very high reliability
$0.8 \leq \alpha < 0.9$	High reliability
$0.7 \leq \alpha < 0.8$	Reliable
$0.5 \leq \alpha < 0.7$	Low reliability
$\alpha < 0.5$	Very low reliability

Cronbach's alpha has several rating categories to indicate whether the tested questionnaire has good value. The Cronbach's alpha rating category is used to assess both aspects of TAM, namely perceived usefulness and perceived ease of use. Both aspects were assessed with a total of 30 respondents who filled out the questionnaire, 8 statements of perceived usefulness and 10 statements of perceived ease of use on the questionnaire.

Table 4. Reliability results of PU questionnaire

Reliability Statistics		
Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.705	.733	9

The reliability test of the PU questionnaire in Table 4 has a Cronbach's alpha of 0.705. It can therefore be concluded that the statements in the PU questionnaire used for the test are in the reliable category.

Table 5. Reliability results of PEOU questionnaire

Reliability Statistics		
Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.727	.728	10

The reliability test of the PEOU questionnaire in Table 5 has a Cronbach's alpha of 0.727. It can therefore be concluded that the statements in the PEOU questionnaire used for testing are in the reliable category.

Table 6. Questionnaire results

No	Perceived Usefulness								Perceived Ease of Use								
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17

1	5	4	5	4	5	4	5	4	5	4	5	4	5	4	5	4	5	4
2	5	4	5	4	5	4	5	4	5	4	5	4	5	4	5	4	5	4
3	4	4	4	4	4	4	4	4	5	4	5	4	5	4	4	4	5	4
4	5	4	5	4	4	3	4	4	5	4	5	4	5	4	5	4	4	4
5	5	4	5	4	5	4	4	4	5	4	5	4	4	3	4	3	4	3
6	5	4	5	4	5	3	4	3	5	3	5	4	5	4	5	4	5	4
7	5	4	5	3	5	4	5	4	4	3	5	4	5	4	5	4	5	4
8	4	3	4	3	4	3	4	3	5	3	5	3	5	3	5	3	5	3
9	5	3	5	3	5	4	5	4	5	4	5	4	5	4	5	4	5	4
10	5	4	5	4	5	4	4	4	4	3	4	4	5	4	4	4	5	4
11	5	4	5	4	5	3	5	3	5	4	5	4	4	3	5	4	5	3
12	5	4	5	4	5	4	4	4	5	4	5	4	4	4	5	4	5	4
13	4	3	4	4	4	3	4	4	5	3	5	4	4	3	5	4	4	3
14	5	4	5	4	5	4	5	4	4	3	4	4	4	3	4	4	4	4
15	5	4	5	3	4	3	4	3	4	4	4	4	4	3	4	3	4	3
16	5	3	5	3	5	4	5	4	5	4	5	4	5	3	4	3	4	3
17	5	4	5	4	5	4	4	4	5	4	5	4	4	4	5	4	5	4
18	5	3	5	3	5	4	5	4	5	4	5	4	5	4	5	4	5	4
19	5	3	5	3	4	3	4	3	4	3	4	4	4	4	4	3	4	3
20	5	4	5	4	5	4	4	4	5	3	4	4	4	4	5	3	5	4
21	5	3	4	4	5	3	5	4	5	4	5	4	4	4	5	4	4	4
22	5	3	5	3	4	3	5	4	4	3	4	3	4	4	4	4	5	4
23	5	4	5	4	4	4	5	4	5	4	4	4	5	4	5	4	4	3
24	5	4	5	4	5	4	5	3	5	3	5	4	4	4	5	4	5	3
25	5	4	5	4	4	3	4	4	5	4	4	4	5	4	4	4	4	3
26	5	4	5	4	4	3	4	3	5	4	5	4	4	4	20	4	4	3
27	5	4	5	4	5	4	5	4	4	3	4	3	4	4	4	4	4	3
28	5	3	5	3	5	4	5	4	5	4	4	4	4	3	4	4	5	4
29	5	4	5	4	5	4	5	3	5	4	5	4	5	4	4	4	4	3
30	4	3	4	3	5	4	5	4	4	3	4	3	4	3	5	4	5	4

The questionnaire results in Table 6 have different values from 30 student respondents regarding the perceived usefulness after testing Opacity Highlights and Outline Borders. Components Q1 and Q2 cover the speed of users in completing the assigned tasks, components Q3 and Q4 cover the effectiveness of users in analyzing the assigned tasks, components Q5, Q6, Q7, and Q8 cover the accuracy of users in the process of understanding assurance cases and reducing errors in modeling the assigned tasks.

Related to perceived ease of use after testing opacity highlights and outline border. Components Q9 and Q10 cover the ease of understanding assurance cases, components Q11 and Q12 cover the ease of grouping assurance case model elements, components Q13 and Q14 cover the ease of learning how to use opacity highlights and outline borders, components Q15 and Q16 cover the ease of access to grouping assurance case elements, components Q17 and Q18 cover the ease of selecting assurance case elements in the SACMN application. Thus, it can be concluded that the statements in the questionnaire results about aspects of perceived usefulness and perceived ease of use that have a value of 5 are included in the strongly agree criteria and a value of 4 is included in the agree criteria.

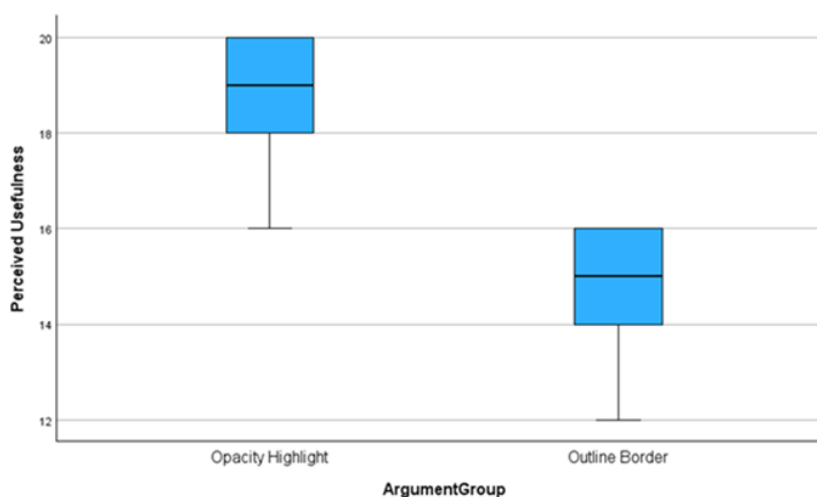


Figure 7. Test result of ArgumentGroup aspect PU

The ArgumentGroup PU test in Figure 7 has a PU value for Opacity Highlights of 20 highest values and 16 lowest values, while the PU value for Outline Borders is 16 highest values and 12 lowest values. It can be concluded that the opacity highlight in the ArgumentGroup PU aspect used for testing is better than the outline border.

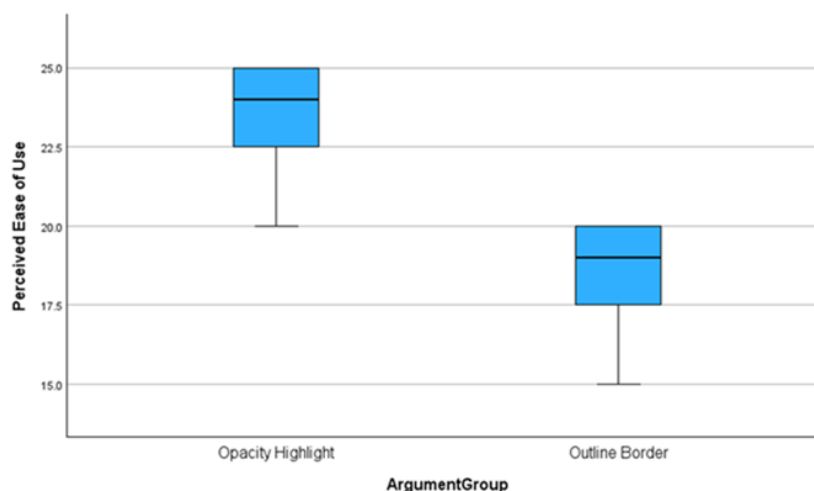


Figure 8. Test result of ArgumentGroup aspect PEOU

The PEOU aspect of the ArgumentGroup test in Figure 8 has a PEOU value for opacity highlight of 25 highest and 20 lowest values, while the PEOU value for outline border is 20 highest and 17 lowest values. From this, it can be concluded that the opacity highlight in the ArgumentGroup PEOU aspect used for testing is better than the outline border.

All users who tested the outline border and opacity highlight are new to ArgumentGroup, but some users become more curious about ArgumentGroup after testing, so an analysis is performed based on the data from 30 respondents. The reliability test results of the PU and PEOU questionnaires are in the reliable category and can be used as a consistent and stable questionnaire.

The perceived usefulness and perceived ease of use questionnaires that have been answered by the respondents, the results of the perceived usefulness questionnaire on the opacity highlight statement in Q1, Q3, Q5 and Q7 are higher than the outline border in Q2, Q4, Q6 and Q8 that all users feel the benefits of using opacity highlights are better in terms of speed, effectiveness and accuracy in completing the assigned tasks than using outline borders.

The results of the perceived ease of use questionnaire on the opacity highlight statement in Q9, Q11, Q13, Q15, and Q17 are higher than the outline border in Q10, Q12, Q14, Q16, and Q18 that all users find it easier to use opacity highlights in terms of understanding assurance cases, accessing grouping elements, and learning the process of selecting assurance case elements than using outline borders in the SACMN application.

4. CONCLUSION

Based on the results of testing and analysis that has been done, it can be concluded that opacity highlights can be used as a representation of the SACMN ArgumentGroup and this application is successfully implemented using the SACMN Case Assurance Model. The results of the questionnaire given and tested the two aspects of TAM, namely perceived usefulness (PU) and perceived ease of use (PEOU) using the assessment method of Cronbach's alpha and get reliable results with 0.705 and 0.727 which are included in the "reliable" category and it can be concluded that the data from this questionnaire is consistent and stable to be used as a guide in developing applications. From the data shown in the ArgumentGroup test results of PU and PEOU aspects, both aspects of opacity highlighting are better in terms of speed, effectiveness and ease of completing the assigned tasks and get good scores and can be categorised as successful or strongly agree with user perceptions (with the highest scores of 25 and 20). Suggestions for further research, it is hoped that there will be further development of existing applications and can try to improve some of the code in the application relating to SACMN elements and ArgumentGroups so that the use of applications can be more comfortable, understandable and facilitate the flow in using the application. A wider collection of questionnaire data and the use of opacity-based graphical highlighting can be developed for the notation of other elements in each SACMN assurance case model so that it becomes one of the references for solving assurance cases that are more complex than the current one.

REFERENCES

- [1] Omg, "An OMG® Structured Assurance Case Metamodel TM Publication Structured Assurance Case Metamodel (SACM) Version 2.2," 2021. [Online]. Available: <https://www.omg.org/spec/SACM>.
- [2] R. Hawkins, I. Habli, D. Kolovos, R. Paige, and T. Kelly, "Weaving an Assurance Case from Design: A Model-Based Approach," in Proceedings of IEEE International Symposium on High Assurance Systems Engineering, IEEE Computer Society, Jan. 2015, pp. 110–117. doi: 10.1109/HASE.2015.25.

- [3] M. Maksimov, N. L. S. Fung, S. Kokaly, and M. Chechik, "Two decades of assurance case tools: A survey," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2018, pp. 49–59. doi: 10.1007/978-3-319-99229-7_6.
- [4] R. Wei, T. P. Kelly, X. Dai, S. Zhao, and R. Hawkins, "Model based system assurance using the structured assurance case metamodel," *Journal of Systems and Software*, vol. 154, pp. 211–233, Aug. 2019, doi: 10.1016/j.jss.2019.05.013.
- [5] N. Selviandro, R. Hawkins, and I. Habli, "A Visual Notation for the Representation of Assurance Cases Using SACM," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, 2020, pp. 3–18. doi: 10.1007/978-3-030-58920-2_1.
- [6] N. Selviandro, "Metamodel-Driven Development of an Assurance Case Notation," 2021.
- [7] C. Picardi, R. Hawkins, C. Paterson, and I. Habli, "A Pattern for Arguing the Assurance of Machine Learning in Medical Diagnosis Systems," in *International Conference on Computer Safety, Reliability, and Security SAFECOMP 2019: Computer Safety, Reliability, and Security*, Springer, Cham, Aug. 2019.
- [8] C. Picardi and I. Habli, "Perspectives on Assurance Case Development for Retinal Disease Diagnosis Using Deep Learning," in *AIME 2019: Artificial Intelligence in Medicine*, Springer, Cham, May 2019, pp. 365–370.
- [9] G. Jošt, M. Heričko, and G. Polančič, "Theoretical foundations and implementation of business process diagrams' complexity management technique based on highlights," *Softw Syst Model*, vol. 18, no. 2, pp. 1079–1095, Apr. 2019, doi: 10.1007/s10270-017-0618-5.
- [10] G. Jošt, J. Huber, M. Heričko, and G. Polančič, "Improving cognitive effectiveness of business process diagrams with opacity-driven graphical highlights," *Decis Support Syst*, vol. 103, pp. 58–69, Nov. 2017, doi: 10.1016/j.dss.2017.09.003.
- [11] R. A. Sutopo, N. Selviandro, and G. S. Wulandari, "Analysis and Implementation of Web-based Graphic Editor for Structured Assurance Case Metamodel Notation," in *1st International Conference on Software Engineering and Information Technology (ICoSEIT)*, 2022, pp. 222–227.
- [12] M. Salahshour Rad, M. Nilashi, and H. Mohamed Dahlan, "Information technology adoption: a review of the literature and classification," *Universal Access in the Information Society*, vol. 17, no. 2. Springer Verlag, pp. 361–390, Jun. 01, 2018. doi: 10.1007/s10209-017-0534-z.
- [13] H. Taherdoost, "A review of technology acceptance and adoption models and theories," in *Procedia Manufacturing*, Elsevier B.V., 2018, pp. 960–967. doi: 10.1016/j.promfg.2018.03.137.
- [14] P. Lai, "The literature review of technology adoption models and theories for the novelty technology," *Journal of Information Systems and Technology Management*, vol. 14, no. 1, pp. 21–38, Jun. 2017, doi: 10.4301/S1807-17752017000100002.
- [15] E. Denney and G. Pai, "Tool support for assurance case development," *Automated Software Engineering*, vol. 25, no. 3, pp. 435–499, Sep. 2018, doi: 10.1007/s10515-017-0230-5.
- [16] S. Fortmann-Roe, "Insight Maker: A general-purpose tool for web-based modeling & simulation," *Simul Model Pract Theory*, vol. 47, pp. 28–45, 2014, doi: 10.1016/j.simpat.2014.03.013.
- [17] P. J. Graydon and C. M. Holloway, "An investigation of proposed techniques for quantifying confidence in assurance arguments," *Safety Science*, vol. 92. Elsevier B.V., pp. 53–65, Feb. 01, 2017. doi: 10.1016/j.ssci.2016.09.014.
- [18] O. Jaradat, I. Sljivo, I. Habli, and R. Hawkins, "Challenges of Safety Assurance for Industry 4.0," in *Proceedings - 2017 13th European Dependable Computing Conference, EDCC 2017*, Institute of Electrical and Electronics Engineers Inc., Nov. 2017, pp. 103–106. doi: 10.1109/EDCC.2017.21.
- [19] K. Tuma, G. Calikli, and R. Scandariato, "Threat analysis of software systems: A systematic literature review," *Journal of Systems and Software*, vol. 144, pp. 275–294, Oct. 2018, doi: 10.1016/j.jss.2018.06.073.