computing@computingonline.net
www.computingonline.net

Print ISSN 1727-6209
On-line ISSN 2312-5381
International Journal of Computing

ASSURANCE CASE FOR SAFETY AND SECURITY IMPLEMENTATION: A SURVEY OF APPLICATIONS

Vladimir Sklyar, Vyacheslav Kharchenko

Dep. 503, National Aerospace University "KhAI", Chkalov str., 17, Kharkiv, 61070, Ukraine,
{v.sklyar, v.kharchenko}@csn.khai.edu

Paper history:
Received 12 May 2020
Received in revised form 8 July 2020
Accepted 17 October 2020
Available online 30 December 2020

Keywords:
Assurance Case;
safety;
security;
dependability.

Abstract: This paper presents a survey of Assurance Case implementation for applications which are not directly related to the usual for Assurance Case regulatory regime. The UK is the country which first developed the theory of Assurance Case as a response to big catastrophes, and most applies Assurance Case regime for many industrial domains. USA, Australia and EU countries apply Assurance Case approach for safety and security regulation and licensing. For the last two decades Assurance Case has been used mostly for confirmation analysis of critical systems with established set of regulatory requirements. There are proven standards of use, notations and tools to support Assurance Case methodology. However, many researchers have tried to find approach to expand Assurance Case application to communicating domains. We group the following directions of Assurance Case applications as the following ones: Assurance Case for attributes assessment such as quality, dependability and, first of all, safety and security, Assurance Case based certification, improvement of argumentation, assurance based development, and Assurance Case for knowledge management. The main challenges and solutions of development and application of Assurance Case methodology, techniques and tools have been analyzed.

1. INTRODUCTION

Assurance Case (AC) is a reasoned and compelling argument, supported by a body of evidence, that a system, service or organization will operate as intended for a defined application in a defined environment [1].

For the last two decades AC has become a powerful tool to analyze confirmation of critical systems with established set of requirements. Systems and infrastructures which are analyzed using AC are usually large, complex, risk-intensive and nowadays software-intensive. Regulatory authorities and other stakeholders of licensing processes recognize benefits of AC implementation, since AC increases the depth of study by gathering evidence of safety and security from a range of sources including risk assessments, incident reporting, human factors analysis and operational experience. The AC regime is a means of establishing a formal structure for safety and security related activities and ensuring that a disciplined and standardized approach to managing risk is adopted. Further benefits of AC include [2]: integrating evidence sources, aiding communication among stakeholders, making the implicit issues explicit, aiding management and governance.

In many critical industries, such as nuclear energy, aviation, defense, etc., AC (or Safety Case) is required by the regulator to establish the safety and security of systems or activities. The UK is the country which first developed the theory of AC as a response to big catastrophes and most applies AC regime for many industries. The USA, Australia, as well as European Union develop and apply AC approach to safety and security regulation and licensing.

Implementation of AC for safety regulation is obvious, and there are some detailed researches

This paper has been submitted for the Open Special Issue on Green Mobile Computing and IoT Systems. Assessment, Modeling, Assurance.

which provide well described industrial cases, for example [2]. However, it is worth considering additional application of AC. In this paper we study, how AC can add value to other activities and applications, which are only partly related to regulatory issues.

2. BASIC CONCEPTS

2.1 A BRIEF HISTORY OF ASSURANCE CASE

The historical and theoretical origins of the Assurance Case refer to the field of logical reasoning, such as operations with logical predicates, including the implication. In 1958, the British philosopher Stephen Toulmin published the book “The Uses of Argument” [3], in which he expanded the operation of logical inference with the degree of confidence and additional arguments and counter-arguments. In addition, Toulmin proposed to present the argument in graphical form, and this approach has since become widespread.

At the same time, after the Second World War, the rapid development of complex industries, such as nuclear energy, space technology, oil and gas, chemical industries, and transport began. All this was accompanied by the introduction of new at that time automation technology. As a result, humanity was faced with man-made disasters of unprecedented scale. Also, in the post-war world, human life was recognized as the highest value. The level of acceptable techno-genic risk was set by law at a fairly hard-to-reach level of 10^{-6} 1/year, i.e. one death per million people per year from technical risks [2, 4].

Thus, the predecessor of the AC is historically the Safety Case. The concept of the Safety Case originated in the 1950s, although the term itself appeared later. The first regulatory document requiring the development of a Safety Case for hazardous industrial facilities is the European Union’s “CIMAH (Control of Major Accidents Hazards) Regulations”. The widespread introduction of the Safety Case into practice began to occur after an unprecedented accident on the Piper Alpha oil platform in the North Sea, which claimed the lives of 167 people in 1988 [4].

All of the above has led to new approaches in safety assessment and assurance. In the 1990s, Toulmin’s argument was used as the basis for the development of semi-formal notations to justify safety [5]. The work was done in the UK, at the University of York, where Goal Structuring Notation (GSN) was developed [1]. Adelard, that is a British company dealing with safety, security and risk management, developed the Claim, Argument and Evidence (CAE) notation in parallel [6] with GSN. These two notations are mainly used in the present (see section 2.2).

Initially, the focus was on functional safety issues, that was named as Safety Case, then with the advent of the information security problem, a similar approach was extended to the Security Case (or Trustworthiness Case), and with it came the understanding that it was necessary to work simultaneously on providing both safety and security features [7]. Currently, the term Assurance Case means the justification of both safety and security [8].

2.2 ASSURANCE CASE STANDARDS, NOTATIONS AND TOOLS

Different normative documents have been developed to regulate the use of the Assurance Case in the nuclear power industry, aviation, the automotive industry, etc. The most general provisions for the application of the Assurance Case relating to system and software engineering are given in the standards of the ISO/IEC 15026 series “Systems and software engineering – Systems and software assurance” [9], which includes four parts:

– Part 1: Concepts and vocabulary;
– Part 2: Assurance case;
– Part 3: System integrity levels;
– Part 4: Assurance in the life cycle.

Object Management Group (OMG) developed Structured Assurance Case Metamodel (SACM) [7]. Goal Structured Notation Community Standard [1] is closely related with OMG SACM providing the GSN description.

ISO 26262:2011 “Road vehicles – Functional safety” standard [10] in ten parts recommends Safety Case implementation for automotive systems. Also some more national and European regulations related with AC are discussed in [2].

Concerning AC notations we give hire only some basic information since there are a lot of papers and reports with very detailed descriptions. At the present there are two the most used AC notations including CAE and GSN.

The CAE notation operates with three specified entities: claim indicates the achievement of the required system properties, evidence provides a documented basis for argumentation, demonstrating the achievement or non-achievement of goals, and arguments are built using inference rules and link evidence with objectives. Arguments such as deterministic (or logical), probabilistic, and qualitative are commonly used. To designate claims, arguments and evidence, graphic primitives with different shapes are introduced [11].

Usually CAE notation is applied in graphical view, but tabular view can also be used. In this case, claim, argument and evidence should be located respectively in the fields of the table [12]. Modification of CAE notation was proposed to manage and evaluate safety and security at all stages of the life cycle [13].

Argumentation strategy may be supported by compliance criterion and coverage criterion. Compliance criterion clarifies how compliance with requirement and claim can be achieved. Coverage criterion applies to multiple hierarchical requirements (for example, when all requirements must be verified during the testing process). Thus, CAE notation is transformed into CAEC notation (Claim, Argument, Evidence and Criteria) [14].

GSN [1], like CAE, operates with entities such as goal (analogous to claim), argumentation strategy, and a solution (analogous to evidence). GSN introduces the context, which is used for informational support of goal setting. Assumptions and justifications also are parts of GSN and can be used to support argumentation. The goal structure is also hierarchical.

Today, there are three of the most functional software tools that are used to create and maintain the Assurance Case [15]. All of them have a paid license.

The first and the most widely used tool is the ASCE (Assurance and Safety Case Environment), which has been developed and maintained by the British company Adelard since the 1990s [16]. In the UK, the development of the Assurance Case is required by laws and standards in many areas related to safety and security, so ASCE has a fairly large market there. Adelard ASCE supports both CAE and GSN. The main part of the tool is a graphic editor, in which additional text or hyperlink information may be attached to graphic blocks.

The next software tool is Astah GSN developed by Change Vision Company from Japan [17]. The company was created in 2006. Astah GSN was developed as a part of the Astah Professional toolkit, which is a media for complex systems modeling. As the name suggests, this program supports only GSN. In addition, it can create Mind Map diagrams. In the graphical editor, you can attach text and hyperlinks to graphic symbols.

The software tool NOR-STA was developed by the Polish company Argevide, which was founded by the staff of the University of Gdansk. NOR-STA supports its own TRUST-IT notation [18], which complies with the provisions of the standard ISO/IEC 15026. The difference is that, instead of a graphical representation, the NOR-STA uses a structured hierarchical list.

Entities in hierarchical Assurance Case list are indicated by different icons. To confirm compliance with the claim, the argumentation strategy is used, and facts or observations, rationale, assumptions and sub-claims are used as analogue of the evidence. Unlike the two previous desktop applications, NOR-STA is used online and supports distributed team work.

Additionally, NASA Ames Research Center reported development a toolset AdvoCATE for assurance case automation [19].

Analyzed standards, tools, used cases and applications areas demonstrate, that, on the one hand, AC methodology has more potential than only safety and security justification. This potential consists of some additional aspects of AC application which can support safety and security assessment. On the other hand, AC requests explicit investigation and description of these additional applications. The objective of this paper is to perform a survey of the AC applications to critically analyze existing advantages and drawbacks. We use applicable scientific articles and works to perform this survey. We identify the following areas of AC applications:

– AC for attributes assessment (see Section 3);

– AC based certification (see Section 4);

– Improvement of argumentation (see Section 5);

– Assurance based development (see Section 6);

– AC for knowledge management (see Section 7).

3. ASSURANCE CASE FOR ATTRIBUTES ASSESSMENT

Some works in 2000s extended the concept of AC to the higher level of system attributes. This AC application was developed by a research group from Software Engineering Institute of Carnegie Mellon University (CMU/SEI). The report [20] discusses Dependability Case for communication system using GSN. It is only terminological issue because an approach is identical with AC.

An idea is, if only dependability or quality attribute of interest is safety, then Dependability or Quality Case becomes Safety Case [20]. The same is right for Security Case which can be a particular case of Dependability or Quality Case. This also entails a general concept of AC which can be an umbrella for different system attributes including dependability, quality, safety and security. Fig. 1 shows dependencies between levels of attributes and associated cases.

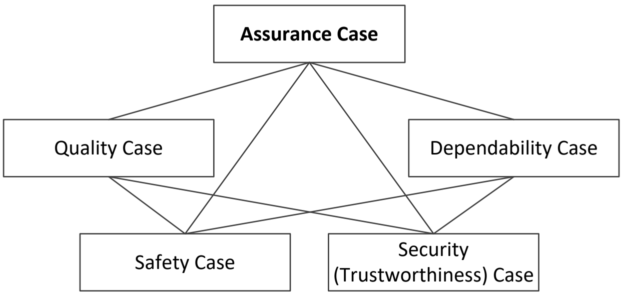

Figure 1 – Relations between cases of different attributes sets

So, Quality Case and Dependability Case are particular cases of Assurance Case, as well as Safety Case and Security Case are particular cases of Assurance Case, Quality Case and Dependability Case. It is worth noting that nowadays terms “Quality Case” and “Dependability Case” are not widely used. There is not any difference in methodology between AC and any other kind of “case”. Also the Nimrod Review [21] recommended that Safety Cases should be renamed as “Risk Cases”. The recommendation to rename Safety Cases has not been adopted by the UK Ministry of Defense.

The CMU/SEI also proposed the Survivability Analysis Framework (SAF) that is a structured view of people, process, and technology that was developed to help organizations characterize the complexity of multi-system and multi-organizational business processes [22]. By combining SAF and GSN based AC, the strengths and gaps for the survivability of a business process can be described in a graphical and visually compelling form that management, architects, system engineers, software engineers, and users can share.

The paper [23] considers AC as a tool implementing integral software assurance to reduce risks and ensure that the software is dependable and trustworthy.

The handbook [24] represents an approach to assess quality of software-based systems with use of AC methodology. The proposed Quality Case is based on CAE entities, but the used notation is derived from the Unified Modeling Language (UML) class diagram that makes it simplified from usual CAE notation (see Fig. 2).

Security Case and Safety Case in [24] are considered as parts of Quality Case as soon as Quality is stated as the top level property. Quality of systems is associated with quality of system architecture, so Architecture Quality Case should be developed as a part of general Quality Case. The handbook [24] contains a lot of examples of Quality Case parts. A general method is named as QUASAR and this method includes:

– teams and member roles with associated responsibilities;

– four phases (Architecture Assessment Initiation, Requirements Review, Architecture Assessment, Architecture Assessment Summary) consisting of associated tasks and component steps;

– work products that are produced and used by members of these teams during the QUASAR phases and tasks.

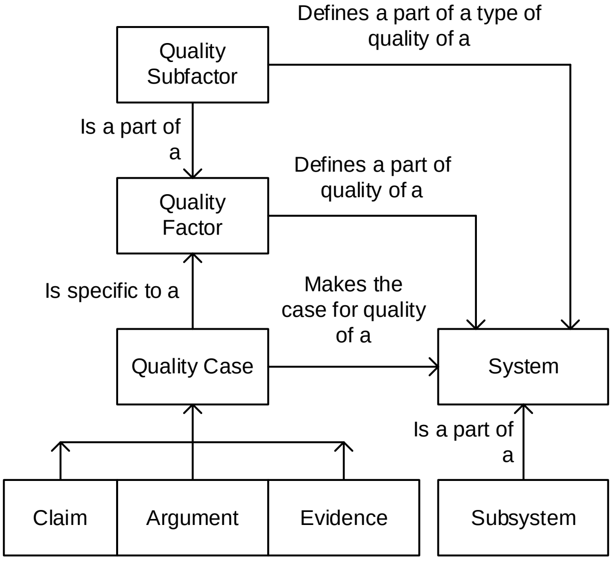

Figure 2 – Structure of Quality Case

The QUASAR method has been implemented by the US Department of Defense (DoD) for many aircraft and ground-based systems, that characterize it as mature and proven-in-use application of AC.

4. ASSURANCE CASE BASED CERTIFICATION

Certification activity is very close to licensing activity [2], so it is obvious, there are researchers efforts directed to application of AC for certification goals. Certification is a process itself which is to substantiate the compliance of applicable requirements with critical software and systems. With the recommended processes which are intended to support certification, it is easy and clear for duty-holders to organize and plan activities and resources in the development lifecycle. The main idea is integration of AC regime with existing regulation and practice in certification. Practical guidance is required to formulate arguments, appropriately select evidence and perform critically review for AC [25].

One from the first research, proposed to extract requirements from standards for AC building need, was published as [26]. The paper [26] is devoted to mapping AC from three standards:

– ISO/IEC 15408 Information technology – Security techniques – Evaluation criteria for IT security (The Common Criteria) [27];

– RTCA/DO-178 Software Considerations in Airborne Systems and Equipment Certification [28];

– ISO 14971 Medical devices – Application of risk management to medical devices [29].

The paper [30] is also devoted to provide argumentation on the base of the perspective Common Criteria (the standard ISO/IEC 15408). The above provides the basis to other industries specific researches, for example for civil aircrafts which is not covered with AC requirements and methodology [25].

The paper [31] provides results of development of as named explicit "e78-1.6" Assurance Case, which is intended to properly capture that is required by the avionic standard RTCA DO-178 [28]. So AC may help serve as a catalyst for prompting improved cooperation and mutual understanding between supporters of prescriptive standards and supporters of goal-based standards. However, the decision to implement AC for civil aircrafts has not been made by now.

In the paper [32] authors presented an approach to certification that provides an operational definition of the various required attributes which are otherwise undefined ("compelling," and "valid"). Satisfactory completion of the certification process implies that the associated AC has those attributes. This operational definition is testable and provides both certifier and applicant with practical engineering goals.

5. IMPROVEMENT OF ARGUMENTATION

A new wave of AC researches appeared after some critical notes were made and named Nimrod Report [21] published in 2009. It became clear, that neither the philosophy literature nor other disciplines that use argument seem to offer a universal theory of knowledge that is applicable to safety arguments [33]. Any item of evidence could be replaced by further argument. Normative models of informal argumentation do not offer clear guidance on when an argument should cite evidence rather than appeal to a more detailed argument. So, improvement of argumentation stimulated a lot of papers devoted to this issue [34, 35], taking into account that there is not any completed agreement which kind of evidence could be completely sufficient.

Epistemology based approach takes into account the study of the nature of knowledge, justification, and the rationality of belief ("What makes justified beliefs really justified?"). The paper [33] hypothesizes that recognition of a set of rules for what counts as sufficient evidence for a given kind of claim under given circumstances would be effective enough. This set of rules could provide developers, assessors, and regulators with a practical means to make justified decisions about how much detail an argument should have and whether an argument is sufficiently compelling.

Eliminative induction was suggested firstly by Sir Francis Bacon for evaluating confidence in a claim [35, 36]. The idea is confidence in a hypothesis (or claim) increases as reasons for doubting its truth are identified and eliminated (Baconian confidence). The report [36] proposes to visualize eliminations in confidence map, a graphical structure with a specific notation that explicitly shows reasons for doubting the validity of the claims, evidence, and reasoning comprising an argument. In particular, the map shows why these doubts are either eliminated or remain as reasons for reduced confidence (see Fig. 3). Also confidence maps can help to discover sources of unsoundness in arguments, namely, questionable inference rules and weaknesses in proffered evidence. The notions of eliminative argumentation and, in particular, the different kinds of defeaters, provide a helpful way of thinking about how to formulate and evaluate arguments.

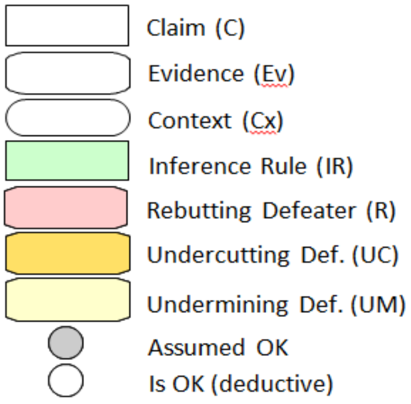

Figure 3 – Confidence Map Symbols

The paper [37] proposes to improve argumentation confidence by converting AC models between different notations. The methods start from argument based cases (CAE or GSN), which are converted into a set of Toulmin model instances; then they use Hitchcock's evaluative criteria [38] for solo verb reasoning to analyze and quantify the Toulmin model instances into Bayesian Belief Network (BBN); running the BBN, quantified confidence from each claim of the AC is got.

The paper [34] surveys how researchers have reasoned uncertainty in assurance cases. The types of uncertainties are addressed and distinguished between qualitative and quantitative approaches. The qualitative approach is covered with Baconian probability [36] and logical argumentation, as in [39].

The paper [39] introduces assured safety arguments. This structure explicitly separates the safety case argument into two components – a safety argument and an accompanying confidence argument. The safety argument is allowed to talk only in terms of the causal chain of risk reduction, and is not allowed to contain general 'confidence raising' arguments.

Quantitative approaches introduce using of probability to define confidence. The paper [40] proposes that probability is the appropriate measure of uncertainty and depended confidence. Researchers explored how the confidence in judgments affects the overall judgment of a safety related probability of failure on demand and illustrated this with an example of Safety Integrity Level (SIL).

In the paper [41] argument structure is presented as for a formal probabilistic treatment of confidence with implementation of the multi-legged approach. For this approach legs of an argument are identical to different versions of argumentations, when more versions give more confidence. It answers questions, for example, such as "How much extra confidence about a system's safety will I have if I add a verification argument leg to an argument leg based upon statistical testing?" There is a simplified and idealized example of a safety system in which interest centers upon a claim about the probability of failure on demand. The approach is based on a BBN model of a two-legged argument, and manipulate this analytically via parameters that define its node probability tables.

6. ASSURANCE BASED DEVELOPMENT

The paper [42] presents Assurance Based Development (ABD) that is an approach to the simultaneous development of systems and their assurance argumentation, which finally shall be represented in a view of AC. ABD ensures that the techniques and means selected to create a system support the correct evidence to justify the required confidence. ABD is based on two key concepts: firstly, engineering choices should be driven by the need to produce evidence for the assurance arguments, and, secondly, argument should be used to document the rationale for believing that the system is fit for use (see Fig. 4).

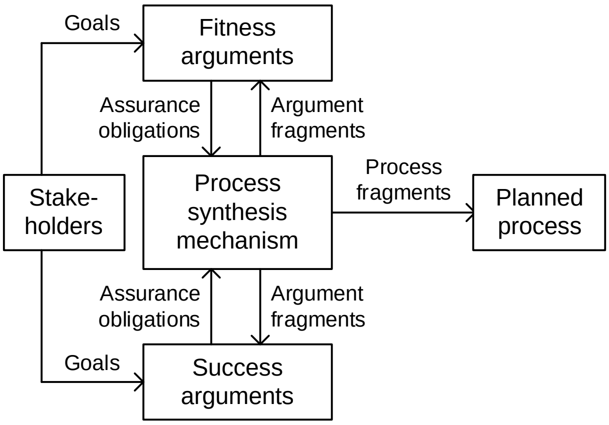

Figure 4 – A concept of Assurance Based Development

Safety contracts method is a modification of approach to ABD, since contracts is an approach to formalize development of software [43]. The paper [44] proposes deriving contracts from fault trees. Such safety contracts guarantee to prevent or minimize the faulty state described by the node. Descriptions of specific safety contracts are implemented in AC diagram as components of GSN. Contract/evidence pairs are represented as C: <A,G>;E, which can be read as follows: contract C, which under assumptions A offers guarantees G, is supported by evidence E.

Another branch of ABD is application of model-based development [8]. The paper [45] is devoted to development of software and AC in parallel following a model-based technique that combines formal modeling of the system, systematic code generation from the formal model, and measurement based verification of timing behavior. The software is developed for an electronic medical device.

In the paper [46] a model-based assurance approach is developed, based on a weaving model, which allows integration between assurance case, design and process models and metamodels (Fig. 5).

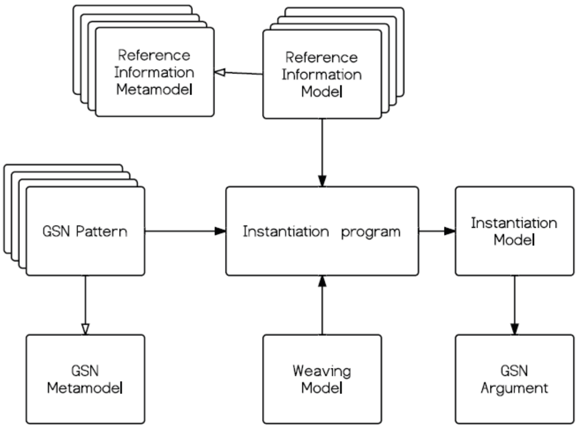

Figure 5 – A model-based Assurance Case

The AC is treated as a structured model, with the aim that all entities in the assurance case become linked explicitly to the models that represent them. The above allows increasing formality and automation of AC. Following a model based development the same authors introduce dynamic AC [47]. This dynamic AC supports a methodology for the systematic engineering of trustworthy self-adaptive software, combining of design-time and runtime modeling and verification with assurance processes to develop trustworthy self-adaptive software and AC arguing the suitability of the software for its intended application.

A system model specified in an Architecture Analysis and Design Language (AADL) is used as an input for AC generation in the paper [48]. Authors stated that the rigor of these automatically generated AC exceeds those of traditional AC arguments because of their more formal logical foundation and direct connection to the architectural model. Authors have implemented special AC language and tool ‘Resolute’ as an AADL annex. ‘Resolute’ embeds the proof in the architectural model, coupling terms in the AC with evidence derived directly from the system design artifacts.

7. ASSURANCE CASE FOR KNOWLEDGE MANAGEMENT

Since AC is a visualization method using a natural language, AC is widely used for support knowledge management and other associated activities such as business management strategy, change and maintenance management, documents management and even software test management.

Some researches in Japan are directed to apply AC methodology for business processes. Kobayashi et al. [49] proposed a method for confirmation and evaluation the management strategy with using AC. The research [49] introduces management vision model, management strategy model, business process model, and IT system model based on AC respectively contribute to improving the feasibility of accomplishing management vision and management strategy.

The paper [50] considers the effectiveness of the advantage of AC as a framework for teaching information security. AC has been used as one of the tools for students during educational project implementation to improve teaching efficiency.

The paper [51] introduces AC to improve testing strategy for space mission critical software. The key step that combines methods is to extract from AC the combinatorial test conditions needed to have confidence in the autonomous system, and feed those conditions into the test suite. This provides an explicit and documented link between the AC and the test generator, which improves confidence and test efficiency. If the AC has to be changed during development, it is easier to update the parameters and re-generate the test cases, thereby reducing regression test costs.

The Japanese Aerospace Exploration Agency (JAXA) implements AC to manage testing activities naming this framework as Independent Verification and Validation (IV&V) case [52]. JAXA identified a range of the following IV&V needs: clear accountability for activities confidence, guarantee of the software quality as a whole, show traceability between software defects on orbit and operational risks. JAXA introduces GSN for sharing and application of knowledge in IV&V area. Obtained effects of IV&V case include improvement of demand and value of IV&V for stakeholders as well as maintenance of IV&V quality.

8. CONCLUSION

We survey AC implementations which directly are not related to regulatory and licensing frameworks for safety critical industries. We found the following areas of AC applications:

− Assurance Case for attributes assessment allows developing Dependability Case, Quality Case, Risk Case, etc. combining different kinds of critical attributes of software, systems and infrastructures. The existing drawback is a presence of different taxonomies for the AC attributes that entails absence of unified approach to the AC structure and content. On the other hand, this feature supports flexible AC development, that can focus on different sets of required attributes. The QUASAR method [24] can be recommended as the most unified and mature from existing approaches to the AC attributes assessment;

− Assurance Case based certification is going to integrate AC regime with existing practices in certification as well as to extract requirements from standards for AC building needs. A practical method has been proposed in [41], but a drawback is in the stated necessity to completely rework existing standards structure to adopt it for explicit AC extraction from the text of standards;

− Improvement of argumentation is directed to improve confidence and to eliminate uncertainty in AC argumentation with qualitative or quantitative approaches. This part essentially affects the AC implementation, since argumentation is a core of the methodology. Drawback of existing researches is absence of agreed and coordinated approach to AC development. We found that structured argumentation [35] can be a solution for any kind of safety and security systems. Also we propose the structured argumentation method supported by structural text and a set of formal operations performed with the AC graph.

− Assurance Based Development considers simultaneous design of systems and their assurance argumentation, which finally shall be represented in a view of AC. This set of engineering techniques is widely used for critical systems. The main proven in use solutions include safety contracts [43] and model-based AC [46];

− Assurance Case for knowledge management supports associated activities such as business management strategy, change and maintenance management, documents management, software test management, etc. This application is the least critical and can be implemented by the manner, which is autonomous from safety and security support. In this case, the implemented method has to be originated from the management needs (see, for example, [49]).

AC is originated from the UK where it is applied in many areas. Concerning other countries AC has not been implemented so widely. Probably it happens because of many challenges in AC application, which are stated, for example in [21]. Besides, additional efforts are needed to assess uncertainties of AC application [53] and decrease influence of them and expert errors on final result by self-assessed case technique.

9. REFERENCES

[1] *GSN Community Standard, Version 1*, Origin Consulting (York) Limited, York, UK, 2011, 64 p.

[2] *Evidence: Using safety cases in industry and healthcare*, Health Foundation, London, UK, 2012, 32 p.

[3] S. Toulmin, *The Uses of Argument*, Cambridge University Press, 1958, 268 p.

[4] W. Cullen, *The Public Enquiry into the Piper Alpha Disaster*, Department of Energy, London, UK, 1990, 488 p.

[5] T. Kelly, *Arguing Safety: A Systematic Approach to Managing Safety Cases. PhD Thesis*, University of York, UK, 1998, 341 p.

[6] *The Adelard Safety Case Development (ASCAD) Manual*, [Online]. Available: http://www.adelard.com/resources/ascad/

[7] *Structured Assurance Case Metamodel, v2.0*, Object Management Group, 2016, 56 p.

[8] R. Wei, T.Kelly, X. Dai, S. Zhao, R. Hawkins, "Model based system assurance using the structured assurance case metamodel," *Journal of Systems and Software*, vol. 154, pp. 211-233, 2019.

[9] *ISO/IEC/IEEE 15026:2019, Systems and software engineering – Systems and software assurance (in 4 parts)*, ISO, Geneva, Switzerland, 2019.

[10] *ISO 26262:2011, Road vehicles – Functional safety*, ISO, Geneva, Switzerland, 2011.

[11] P. Bishop, R. Bloomfield. "A methodology for safety case development," *Safety and Reliability*, vol. 20, issue 1, 2000, pp. 34-42.

[12] V. Sklyar, V. Kharchenko, "Assurance case driven design based on the harmonized framework of safety and security requirements," *Proceedings of the 13th International Conference on ICT in Education, Research and Industrial Applications*, Kyiv, Ukraine, May 15-18, 2017, pp. 670-685.

[13] V. Kharchenko, V. Sklyar. "Assurance case driven design for software and hardware description language based systems," *Radioelectronic and Computer Systems*, no. 5(79), 2016, pp. 98-103.

[14] V. Sklyar, V. Kharchenko, "Green assurance case: Applications for Internet of Things," in: V. Kharchenko, Y. Kondratenko, J. Kacprzyk (Eds.), *Green IT Engineering: Social, Business and Industrial Applications*, Springer, 2019, pp. 351-371.

[15] M. Maksimov, N. Fung, S. Kokaly, M. Chechik, "Two decades of assurance case tools: A survey," in: B. Gallina, A.Skavhaug, E. Schoitsch, F. Bitsch (Eds.), *Computer Safety, Reliability, and Security*, Springer, 2018, pp. 49-59.

[16] *Adelard ASCE Software*, [Online]. Available at: https://www.adelard.com/asce/choosing-asce/index/

[17] *Astah GSN Editor Overview*, [Online]. Available at: http://astah.net/editions/gsn

[18] *NOR-STA: Support for Achieving and Assessing Conformance to NORms and STAndards*, [Online]. Available at: https://www.nor-sta.eu/en

[19] E. Denney, G. Pai, *Tool Support for Assurance Case Development. ARC-E-DAA-TN48294*, NASA Ames Research Center, Moffett Field, CA, USA, 64 p.

[20] C. Weinstock, J. Goodenough, J. Hudak, *Dependability Cases. CMU/SEI-2004-TN-016*, SEI/CMU, Pittsburgh, PA, USA, 2004, 30 p.

[21] C. Haddon-Cave, *The Nimrod Review. An independent review into the broader issues surrounding the loss of the RAF Nimrod MR2 Aircraft XV230 in Afghanistan in 2006*, London, UK, Crown Copyright, 2009, 585 p.

[22] R. Ellison, J. Goodenough, C. Weinstock, C. Woody, *Survivability Assurance for System of Systems. Technical Report CMU/SEI-2008-TR-008*, CMU/SEI, Pittsburgh, PA, USA, 2008, 63 p.

[23] T. Rhodes, F. Boland, E. Fong, M. Kass. "Software assurance using structured assurance case models," *Journal of Research of the NIST*, vol. 115, issue 3, pp. 209-216, 2010.

[24] D. Firesmith, P. Capell, J. Elm, M. Gagliardi, T. Morrow, L. Roush, L. Sha, *QUASAR: A Method for the Quality Assessment of Software-Intensive System Architectures, CMU/SEI-2006-HB-002*, CMU/SEI, Pittsburgh, PA, USA, 2006, 266 p.

[25] L. Sun, W. Zhang, T. Kelly, "Do safety cases have a role in aircraft certification?" *Procedia Engineering*, vol. 17, pp. 358-368, 2011.

[26] T. Ankrum and A. Kromholz, "Structured assurance cases: Three common standards," *Proceedings of the 9th IEEE International Symposium on High-Assurance Systems Engineering*, Heidelberg, Germany October 12-14, 2005, pp. 99-108.

[27] *ISO/IEC 15408:2009 Information technology – Security techniques – Evaluation criteria for IT security*, ISO, Geneva, Switzerland, 2009.

[28] *RTCA/DO-178 Software Considerations in Airborne Systems and Equipment Certification*, RTCA, Washington DC, 2011.

[29] *ISO 14971:2007 Medical devices – Application of risk management to medical devices*, ISO, Geneva, Switzerland, 2007.

[30] R. Hawkins, I. Habli, T. Kelly, J. McDermid, "Assurance cases and prescriptive software safety certification: A comparative study," *Safety Science*, vol. 59, 2013, pp. 55–71.

[31] M. Holloway, "Explicate '78: Uncovering the Implicit Assurance Case in DO–178C," *Proceedings of the 23rd Safety-Critical Systems Symposium*, Bristol, UK, February 3-5, 2015.

[32] P. Graydon, J. Knight and M. Green, "Certification and Safety Cases," *Proceedings of the 28th International Systems Safety Conference*, Minneapolis, MN USA, August 30 – 3 September 04, 2010, pp. 235-244.

[33] P. Graydon and C. Holloway, "Evidence under a Magnifying Glass: Thoughts on Safety Argument Epistemology," *Proceedings of the 10th IET System Safety and Cyber-Security Conference*, Bristol, UK, October 21-22, 2015.

[34] L. Duan, S. Rayadurgam, M. Heimdahl, A. Ayoub, O. Sokolsky, I. Lee, "Reasoning about confidence and uncertainty in assurance cases: A survey," in: M. Huhn, L. Williams (Eds) *Software Engineering in Health Care*, Springer, 2017, pp. 64-80.

[35] J. Rushby. *The Interpretation and Evaluation of Assurance Cases. Technical Report SRI-CSL-15-01*, SRI International, Menlo Park, CA, USA, 2015, 127 p.

[36] J. Goodenough, C. Weinstock, A. Klein. *Eliminative Argumentation: A Basis for Arguing Confidence in System Properties February, Technical Report, CMU/SEI-2015-TR-005*, CMU/SEI, Pittsburgh, PA, USA, 2015, 71 p.

[37] X. Zhao, D. Zhang, M. Lu, F. Zeng, "A new approach to assessment of confidence in assurance cases," in: F. Ortmeier, P. Daniel (Eds.), Computer Safety, *Reliability and Security*, Springer, 2012, pp. 79–91.

[38] D. Hitchcock. "good reasoning on the Toulmin model," *Argumentation*, vol. 19, issue 3, pp. 373–391, 2005.

[39] R. Hawkins, T. Kelly, J. Knight, P. Graydon, "A new approach to creating clear safety arguments," *Proceedings of the 19th Safety Critical Systems Symposium*, Southampton, UK, February 8-10, 2011, pp. 3-23.

[40] R. Bloomfield, B. Littlewood and D. Wright, "Confidence: Its role in dependability cases for risk assessment," in: *Proceedings of the International Conference on Dependable Systems and Networks*, Edinburgh, UK, June 25-28, 2007, pp. 338–346.

[41] B. Littlewood, D. Wrigh. "The use of multilegged arguments to increase confidence in safety claims for software-based systems: A study based on a BBN analysis of an idealized example," *IEEE Transactions on Software Engineering*, vol. 33, issue 5, pp. 347–365, 2007.

[42] P. Graydon, J. Knight. *Assurance Based Development. Technical Report CS-2009-10*, University of Virginia, Charlottesville, VA, USA, 2009, 43 p.

[43] I. Sljivo, B. Gallina, J. Carlson, H. Hansson, "Generation of safety case argument-fragments from safety contracts," in: A. Bondavalli, F. Di Giandomenico (Eds.), *Computer Safety, Reliability, and Security*, Springer, 2014, pp. 170-185.

[44] I. Sljivo, O. Jaradat, I. Bate and P. Graydon, "Deriving safety contracts to support architecture design of safety critical systems," in: *Proceedings of the 2015 IEEE 16th International Symposium on High Assurance Systems Engineering*, Washington DC, USA, January 08-10, 2015, pp. 126-133.

[45] E. Jee, I. Lee, O. Sokolsky, "Assurance cases in model-driven development of the pacemaker software," in: T. Margaria, B. Steffen (Eds.), *Leveraging Applications of Formal Methods, Verification, and Validation*, Springer, 2010, pp. 343-356.

[46] 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 16th International Symposium on High Assurance Systems Engineering*, Daytona Beach, Florida, USA, January 8-10, 2015, pp. 110-117.

[47] R. Calinescu, S. Gerasimou, I. Habli, M. Iftikhar, T. Kelly, D. Weyns. "Engineering trustworthy self-adaptive software with dynamic assurance cases," *IEEE Transactions on Software Engineering*, vol. 44, issue 11, pp. 1-30, 2018.

[48] A. Gacek, J. Backes, D. Cofer, K. Slind and M. Whalen, "Resolute: An assurance case language for architecture models," *Proceedings of the 2014 ACM SIGAda Annual Conference on High Integrity Language Technology*, Portland, OR, USA, October 18-21, 2014, pp. 19-28.

[49] N. Kobayashi, A. Nakamoto, N. Kawase, F. Sussan, S. Shirasaka. "What model(s) of assurance cases will increase the feasibility of accomplishing both vision and strategy?" *Review of Integrative Business and Economics Research*, vol. 7, issue 2, pp. 1-17, 2018.

[50] R. Gallo, R. Dahab, "Assurance cases as a didactic tool for information security," in: M. Bishop, N. Miloslavskaya, M. Theocharidou (Eds.), *Information Security Education Across the Curriculum*, Springer, 2015, pp. 15-26.

[51] B. Smith, M. Feather and T. Huntsberger, "A hybrid method of assurance cases and testing for improved confidence in autonomous space systems," *Proceedings of the AIAA SciTech 2018 Forum*, Kissimmee, FL, USA, January 8-12, 2018, pp. 1566-1577.

[52] K. Kakimoto, K. Sasaki, H. Umeda, and Y. Ueda, "IV&V case: Empirical study of software independent verification and validation based on safety case," *Proceedings of the 2017 IEEE International Symposium on Software Reliability Engineering Workshops*, Toulouse, France, October 23-26, 2017, pp. 32-35.

[53] O. Illiashenko, O. Potii, and D. Komin, "Advanced security assurance case based on ISO/IEC 15408," *Proceedings of the 10th Conference on Dependability and Complex Systems*, Brunów, Poland, June 29 – July 3 2015, pp. 391-401.

__Vladimir Sklyar__, Dr of Science, Professor, is a Professor within the Department of Computer Systems, Networks and Cyber Security at the National Aerospace University "KhAI" (Kharkiv, Ukraine). He has a PhD from Kharkiv Military University and a MS from the same university. His scientific interests lay in area of the safety and security assessment and assurance.

__Vyacheslav Kharchenko__, Dr of Science, Professor, is the Head of the Department of Computer Systems, Networks and Cyber Security at the National Aerospace University "KhAI" (Kharkiv, Ukraine) and the Head of STC, RPC Radiy. He has a PhD and Engineer Degree in Control Systems from Kharkiv Military University. His scientific interests lay in area of the critical and green computing.