This is a postprint of the paper presented at
the 10th IET System Safety and Cyber Security Conference, 20-22 October 2015, Bristol, UK.

Integrating Confidence and Assurance Arguments

A. Jarzębowicz*†, A. Wardziński*†

* Department of Software Engineering, Faculty of Electronics, Telecommunications and Informatics,
Gdańsk University of Technology, Narutowicza 11/12, 80-233 Gdańsk, Poland
{olek, award}@eti.pg.gda.pl
† Argevide sp. z o.o., Poland
{aleksander.jarzebowicz, andrzej.wardzinski}@argevide.com

Keywords: assurance case, safety case, confidence argument, assurance deficit, defeater.

Abstract

To be considered compelling an assurance case should address its potential deficits, possibly with the use of a confidence argument. Assurance argument and confidence argument should be clearly separated and consistent at the same time. We propose a way of their integration with the use of an element representing rationale for each argumentation strategy. The rationale integrates confidence argument for a given argumentation step and can be used to demonstrate strength of the argument. The approach is illustrated with a confidence argument development case study. The confidence argument has been created for defeaters identified with the use of a checklist.

1 Introduction

An assurance case is a reasoned and compelling argument, supported by a body of evidence, that a system, service or organisation will operate as intended for a defined application in a defined environment [17]. Similar definitions (usually based on the definition of a safety case [6]) can be found in e.g. [1, 22]. All of them stress that an argument expressed by an assurance case is supposed to be compelling and valid.

To be compelling and valid, the assurance case should not only provide direct argument and evidence supporting its claims. It should also avoid argumentation fallacies [15,29] and address its own potential deficits like questionable reasoning, inadequate evidence, or uncertainty in general. Such need is being more widely recognized (e.g. [4, 8, 10, 15, 16, 18, 19, 21, 30]). It is vital for the deficits to be identified and dealt with, instead of being hidden to make an assurance case look better – safety and other critical properties can only be argued by challenging assumptions and doubts, otherwise it's just a "paper exercise" [23]. Some authorities, like FDA, recommend that a separate confidence argument is to be provided as part of an assurance case submission [9].

Our objective is to work out a systematic method of identifying assurance deficits and building the confidence argument. The method includes a way of representing confidence arguments in the context of assurance argument.

2 Defeaters and confidence argument

The concept of the confidence argument is based on defending the assurance argument against any doubts, uncertainties, deficiencies and counterarguments. To ensure completeness one needs a systematic approach to identify all the assurance case deficits.

2.1 Defeaters

A number of approaches of identifying assurance case deficits can be found in the literature. Different names and definitions are used like deficits, defeaters or rebuttals.

Hawkins et al. [19] analysed the structure of assurance arguments developed using Goal Structuring Notation (GSN) [17] and identified three types of assertions related to the sufficiency and appropriateness of GSN elements connections and named them Assurance Claim Points (ACPs). Challenging any of the assertions results in a potential defeater. Each identified defeater requires to be addressed by an additional confidence argument:

- Asserted inferences (ACP1) – demonstrate why it should be believed that the supporting premises are sufficient to draw the conclusion;
- Asserted context (ACP2) – demonstrate why it should be believed that citing the specified context defines the appropriate context at this point in the argument;
- Asserted solution (ACP3) – the assurance of the solution depends upon the confidence that the evidence is trustworthy and appropriate to support the claim.

The alternative approach is based on Toulmin's work on defeasible reasoning and model of arguments [27], shown in Fig. 1. Main Toulmin's model specifies the argument as applying the inference rule (warrant - W) to specific data (premises – P1, P2) to justify the conclusion (claim - C). Additional elements are used to express conditions of argument's validity: backing (B) as an explanation and support for inference rule, rebuttals (R1-R4) as the exceptional conditions invalidating the argument and qualifier as restriction of the strength of the argument e.g. "always", "usually", "possibly" (not depicted). The similarity of the rebuttal to the concept of a defeater is not accidental, these two terms can be used interchangeably.

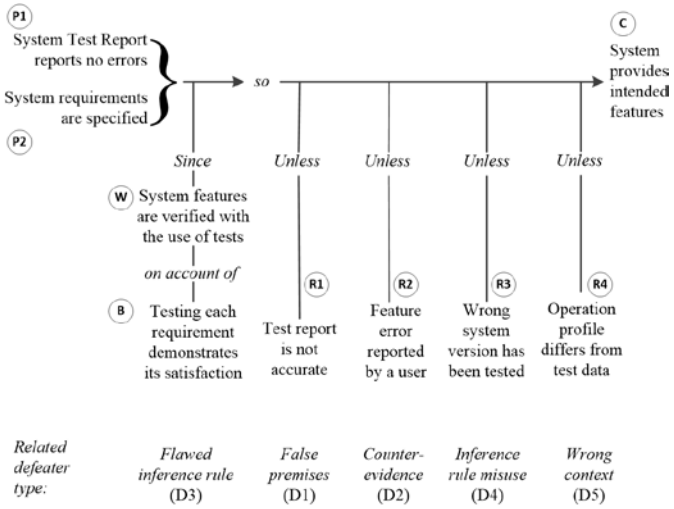

Fig. 1. Toulmin argument example with mapping to defeaters

The first application of defeaters derived from Toulmin's model to assurance cases was proposed by Kelly [21], as part of a multi-staged process of the argument review. The last stage (named "Argument criticism and defeat") includes two forms of defeating an argument:

- Rebuttal – defeating conclusion by providing the information that contradicts it (note that the term Rebuttal used here has narrower meaning than the same term in Toulmin model);
- Undercutting – challenging the inference rule by providing additional information about conditions under which the claim is not necessarily true even if the premises are true.

Goodenough et al. [10] distinguish 3 kinds of defeaters: Rebutting, Undercutting and Undermining. The first two are identical to the Kelly's defeaters [21], while the third one is defined as:

- Undermining – defeater invalidating one or more premises (in which case even if the inference rule is valid and all rebutting defeaters have been eliminated, we still have a reduced basis for believing in the truth of the associated claim).

Grigorova and Maibaum [16] explore further the classification of defeaters applicable to Toulmin's model. They use the classification developed by Verheij [28] as part of the attempt to formalise Toulmin's argument model. As the result five types of defeaters have been listed [16]:

- Providing arguments against the evidence (D1);
- Providing arguments directly against the claim, usually counterevidence (D2);
- Attacking the validity of the inference rule (D3);
- Attacking the connection between premises and the conclusion usually caused by misuse of the inference rule, when the premises are true and the reference rule is valid, but their combined use does not justify the claim (D4);
- Attacking the applicability of the inference rule (D5);

We will use the mnemonics D1-D5 when discussing defeaters in our paper. The list expands the classification by Goodenough et al. [10]: D1 is Undermining defeater, D2 is Rebutting defeater, and Undercutting defeater type is split into three more detailed types: D3, D4 and D5.

The illustration of Toulmin's argument model is presented in Fig. 1. The line of reasoning goes from premises P1 and P2 through the inference rule (warrant W) to the conclusion C. The reasoning will fail when the backing B is false or when any of the rebuttals R1 to R4 is true. We can map the backing and rebuttals to defeater types denoted D1 to D5.

2.2 Confidence argument

Defeaters are a tool for identification and classification of all the assurance case deficiencies and weaknesses, which are then to be addressed by providing a confidence argument. The need for confidence arguments is widely recognised, but a few solutions are available (other than just introducing additional confidence-increasing elements to the assurance case and mixing them with "core" assurance elements).

Hawkins et al. [19] provide a number of reasons why the confidence argument should be created as a separate argument, not mixed with the primary assurance argument. Their approach is to identify Assurance Claim Points and then to develop a separate confidence argument structure addressing them. The top claim of the confidence argument is that the sufficient confidence is demonstrated in assurance argument. It is decomposed according to the ACPs categories. The lower levels of argument relate to particular ACPs.

Ayoub et al. [2] also build a separate confidence argument using the "common characteristics map" structure covering relevant process-based issues. The structure provides guidance what confidence aspects should be addressed.

An alternative approach proposed by Goodenough et al. is to use "confidence maps" [11] - additional graphical diagrams associated with GSN assurance case elements. Confidence maps show defeaters (three types defined in [10]) together with the argument and evidence provided to eliminate them. Interestingly, also inference rules are explicitly modelled in confidence maps (but not included in GSN assurance argument).

3 Proposed approach

We propose an approach for building confidence argument and a method of its representation. We assume the method should:

a) provide a systematic way for confidence argument development;
b) keep clear distinction between the assurance argument and the confidence argument;
c) present a consistent and easy to comprehend view on the relation between the assurance argument and related confidence argument for every argumentation step;
d) provide a tool for managing the scope of the argument presentation (assurance argument, confidence argument and both integrated).

Our approach to confidence argument structure is based on two observations. First, despite that in Toulmin's theory the backing and the rebuttal play different roles, they both can be mapped to defeaters (see Fig. 1). Depending on whether they are satisfied or not, the reasoning step from premises to the claim is valid or not. The difference is that the backing is to be true and the rebuttals are to be false to ensure that the claim is satisfied. Our conclusion is that the confidence argument should address both the backing and the rebuttals.

The second observation is that the confidence argument is strongly context-related. For example a testing method may be effective for one software module but not adequate for another one which uses concurrent programming environment. Therefore, the confidence argument can be analysed only in a precisely defined context of the main assurance argument. We agree that it is useful to create a confidence case separate from the assurance case (possibly starting with a reliable development process), however we regard as equally important to present and verify the confidence argument for each step of the assurance argument.

Our approach refers to the concept of an assurance case element usually named justification of an argumentation strategy. The objective of a justification is to provide "rationale for its inclusion or its phrasing" [17] or "justification for the validity or merit of its method of reasoning" [20]. Justification usually is used to demonstrate the inference rule to be valid and reasonable for a given type of the claim. It is often regarded to be context-free in the sense that it can be applied multiple times for a given type of claims.

Our proposal is to define the justification as *rationale that a given argumentation strategy supported by listed premises justifies cited claim in the specified context*. We will use the name "rationale" in this paper to describe this kind of a justification. We can say that the rationale provides a confidence argument specific for a given argumentation step. Note that the definition of rationale is different from the typical use of the justification element. The rationale definition does not contradict the idea and the role of the justification in the argument, but instead expands it. This approach allows to integrate the assurance argument and the confidence argument in a way presented in Fig. 2.

The approach allows to present a consistent view on assurance and confidence premises for each argumentation step. This does not prevent the creation of a separate and complete confidence argument as links can be used to connect the rationale premises with it. The user can work focusing on the assurance argument or on the confidence part or on the coupled arguments jointly.

The approach can be compared to ACP approach [19] which specifies assertions on GSN elements relationship (edges on diagrams). ACP approach keeps separate confidence and assurance arguments and it makes more difficult to provide a user with an integrated view on both arguments using APCs. The advantage of ACPs is that the assertion type can be easily identified by the location of a small black square on a relationship connecting strategy or context or evidence. This can be mapped to defeaters types. Information on the defeater type is not presented graphically in our approach.

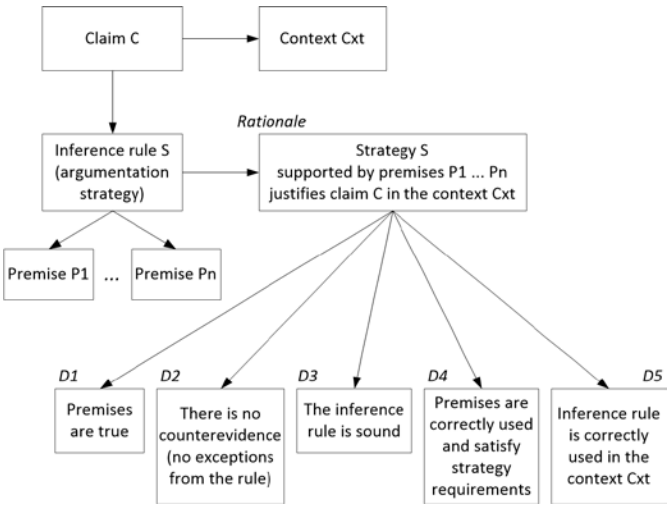

Fig. 2. General schema of argument integration

The main formal difference between our approach and the ACP concept is the role of the justification element of the argument. In our approach we regard it as a root element for a local confidence argument.

In ACP approach each Assurance Claim Point is an assertion providing a reference to the confidence claim, which is a part of the confidence case. In our approach we use links in a similar way. A separate confidence case can be created which contains the overall decomposition of confidence argument. Each of confidence case claim can be connected by a link with a local confidence argument covered by a rationale in the main assurance argument.

Some similarities can be found between our approach and the concept of confidence maps [11]. The main difference is that we integrate assurance and confidence arguments, both having a unified representation, while confidence maps introduce new type of diagrams which explicitly show defeaters and countermeasures for them.

We have used NOR-STA assurance case tool [13] to develop the integrated assurance and confidence arguments. The tool has been developed at Gdańsk University of Technology (GUT) and is using TCL notation (Trust Case Language). TCL is an assurance case notation developed at GUT since 2001 as part of TRUST-IT methodology [12]. The notation is similar to GSN [17] and CAE [6] and is compliant with ISO/IEC 15026 [20] and OMG's SACM [24].

NOR-STA allows to present and filter confidence arguments and to make assessment of assurance case arguments with respect to both assurance and confidence. It uses assessment mechanism based on Dempster-Schafer theory [26, 7]. This allows for assessment aggregation [5, 8, 27] and also for expressing uncertainty instead of giving a choice between "true" and "false" [5, 8].

4 Case study

The presented concept has been implemented in a case study of extending an existing assurance case with a confidence argument. The objectives of the case study were twofold. First to demonstrate the approach for a non-trivial assurance case and also to experience the process of transformation from an ordinary assurance case to the assurance case with the associated confidence arguments.

The case study for a selected assurance case has been conducted in the following steps:

1. For each argumentation step going top-down:
   a. review the argument to identify defeater
2. For each identified defeater:
   a. analyse defeater criticality and decide if it requires actions
   b. identify possible ways for resolution if required
   c. implement resolution:
      – option 1: modify assurance argument structure to eliminate the defeater cause
      – option 2: develop confidence argument to address the defeater
      – option 3: no action required

The argument selected for our case study was the Open PCA Infusion Pump assurance case developed by Kansas State University [25]. The safety part of the assurance case contains over 600 elements, including 96 claims. It covered a broad range of infusion pump hazards and failure modes required by FDA regulations [9], including human aspect, software, hardware and mechanical elements failures. We were aware of the fact that this assurance case is on the concept design level and some of the claims remained undeveloped but nevertheless it suited well for the purpose of the demonstration of our approach.

We planned the review to be conducted by one person, then independently verified by another. To provide the reviewer with some guidance, we decided to use the 5-item list of defeater types (D1-D5) presented in section 2.1. We chose this defeater classification as we considered it to be the most comprehensible one. The reviewer was supposed to immediately assign category (D1-D5) to each defeater. One of the recognised problems related to defeaters is that it is quite easy to raise doubts regarding every piece of information included in the assurance argument and therefore producing a large number of defeaters expressing more and more incredible conditions [11]. We focused on a more practical approach: a defeater was identified only if the reviewer really considered something as a credible doubt e.g. questionable reasoning based on unclear inference rule, missing premises (indicated by inference rule), inadequate hazard mitigation leaving out some causes or conditions.

During the review 127 defeaters have been identified. This was the first such review and it is hard to judge if it is a normal, low or high level. We should also be aware that:

– the reviewed assurance case had the draft status and it was an research case study intentionally not complete (e.g. missing some of the evidence);
– it focused on assurance aspect only, little attention was paid to confidence aspects e.g. related to the process;
– inference rules were seldom explained and justified thus creating reasons for doubt.

Initially the reviewer used the list of defeater types as a checklist, however soon it turned out that free, unguided searching for argument deficiencies was more efficient. The list was too general and difficult to apply to the concrete claims and arguments of the assurance case. Moreover, classification of defeaters, especially those related to inference rules was problematic. For example when the reviewer finds a scenario when the claim is not satisfied, it can be classified as counterevidence (D2). Deeper analysis however may reveal that it is caused by incomplete inference rule for a given context of the claim (D5).

Although it was not planned in our case study, we decided to analyse the defeaters found, try to identify similarities and create a “defeater checklist” using “bottom-up” approach. The resulting checklist is shown in Table 1, together with the percentage share of defeaters found using each checklist item. Quite surprisingly, we found out that grouping the items of the checklist into more general categories resembles 3 types of ACPs [19]. The defeaters we found were either related to the inference rule, premises (evidence but also sub-claims) or the context (e.g. of using the device). The category for each defeater is indicated by the first letter of its identifier.

The next step of our case study was to address a subset of the identified defeaters. The analysed subset covered all the types of defeaters. Some of them were addressed by altering the assurance argument (e.g. refining context), some by developing confidence argument and some by both. The result was the improved assurance case supported by confidence argument.

Let’s consider an example of a claim that given software function works correctly what is argued by module tests where test scope covers function requirements. The argument is presented in thick lines in Fig. 3. The argument review may produce three possible defeaters (however more are possible):

a) premise may be faulty: test report may be incorrect if the testing process is unreliable;
b) argumentation strategy may be faulty: the chosen test method may not be sufficient to provide sufficient confidence that the function works correctly;
c) counterevidence may be available: some function errors may be reported by other tests or operation personnel and not fixed.

Each of the described defeaters can be resolved by adding a confidence claim as presented in Fig. 3.

The main experience from extending the assurance case with supporting confidence argument was that the process involved thorough analysis of the assurance case completeness and consistency. Some of argument deficits were resolved by modification of the main argument without adding any confidence argument.

Some other approaches are focused on building exhaustive confidence argument structure based on trustworthy system

lifecycle and safety management process [2]. In our opinion both directions are important. A reasonable approach would be first to create confidence argument based on the safety management and system development process, then integrate it with the assurance argument for subsequent stages of the assurance case development and periodically review the assurance case to ensure confidence argument completeness.

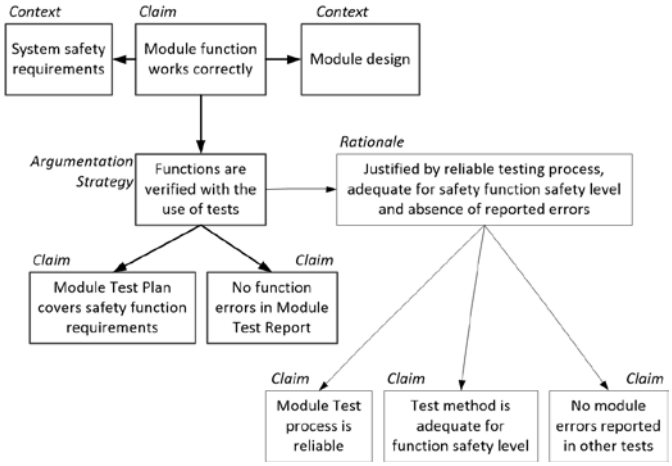

Fig. 3. Confidence argument example

This raises a question how to efficiently build and maintain completeness and consistency of the confidence argument and the main assurance argument. Assurance case review techniques like the checklist presented in Table 1 can be helpful. Moreover, it is not normally achievable to demonstrate complete confidence, therefore a judgement has to be made what level of confidence should be demonstrated for a given part of an assurance case. For example a higher confidence would be required for critical hazards, while for other issues with limited impact on safety a lower confidence level can be accepted and some confidence deficits can be tolerated. We consider that rationale element is suitable to indicate how much confidence is required/demonstrated. Efficient methods for managing the confidence level and consistency between the assurance and confidence argument are needed. Part of the possible solution is the proposed argument integration and the use of reviews and checklists to ensure its completeness.

5 Summary

In this paper we presented an approach for confidence and assurance argument integration and the checklist of defeaters to facilitate the process of building confidence arguments.

The argument integration is based on the idea of extended role of argumentation strategy justification element to represent rationale specific for a given argumentation step. Rationale's objective is to ensure confidence in the argumentation step through decomposition into confidence argument. The approach allows to present comprehensible view of integrated assurance and confidence argument. This does not exclude the construction of separate confidence case as we can use links to connect to "local" confidence arguments.

Confidence argument can be developed using the results of the checklist-based assurance case review. We presented the defeater checklist developed in our case study of a confidence argument development for an Open PCA Infusion Pump assurance case [25]. The checklist can be helpful in maintaining completeness of confidence argument, however it still requires validation with real-life assurance cases. Argumentation reviews and searching for defeaters to develop confidence arguments helps to improve assurance argument quality and eliminate its gaps, ambiguities and weak spots.

We plan to strengthen the process of managing confidence argument completeness and consistency with the assurance argument. Rationale integrates confidence claims necessary to demonstrate the strength of the argumentation step. This can serve as basis for managing the argument strength level depending on assurance argument goals criticality.

References

[1] T. Ankrum, A. Kromholz, "Structured assurance cases: Three common standards", *Proc. of High-Assurance Systems Engineering Symposium (HASE'05)*, Heidelberg, Germany, (2005).

[2] A. Ayoub, B. Kim, I. Lee, O. Sokolsky, "A systematic approach to justifying sufficient confidence in software safety arguments", *Proc. of 31st International Conference on Computer Safety, Reliability and Security (SAFECOMP 2012)*, LNCS 7612, pp. 305-316, (2012).

[3] P. Bishop, R. Bloomfield, "The SHIP Safety Case Approach", *Proc. of 14th International Conference on Computer Safety, Reliability and Security (SAFECOMP'95)*, (1995).

[4] P. Bishop, R. Bloomfield, S. Guerra, "The future of goal-based assurance cases", *Proc. of Workshop on Assurance Cases, 2004 International Conference on Dependable Systems and Networks*, pp. 390-395, (2004).

[5] P. Bishop, R. Bloomfield, B. Littlewood, A. Povyakalo, D. Wright, "Toward a formalism for conservative claims about the dependability of software-based systems", *IEEE Transactions on Software Engineering*, Vol. 37 (2011), pp. 708–717, (2011).

[6] R. Bloomfield, P. Bishop, C. Jones, P. Froome, "ASCAD - Adelard Safety Case Development Manual", Adelard, (1998).

[7] Ł. Cyra, J. Górski, "Support for Argument Structures Review and Assessment", *Reliability Engineering and System Safety*, Vol. 96, Elsevier, pp. 26-37, (2011).

[8] E. Denney, G. Pai, I. Habli, "Towards Measurement of Confidence in Safety Cases", *Proc. of Symposium on Empirical Software Engineering and Measurement*, Banff, Canada, (2011).

[9] US Food and Drug Administration, "Infusion Pumps Total Product Life Cycle. Guidance for Industry and FDA Staff", (2014).

[10] J. B. Goodenough, C. B. Weinstock, A. Z. Klein, "Toward a Theory of Assurance Case Confidence",

Technical Report CMU/SEI 2012 TR 002, Carnegie Mellon University, (2012).
[11] J. B. Goodenough, C. B. Weinstock, A. Z. Klein, "Eliminative induction: A basis for arguing system confidence", *Proc. of 35th International Conference on Software Engineering*, pp. 1161-1164, (2013).
[12] J. Górski, A. Jarzębowicz, R. Leszczyna, J. Miler, M. Olszewski, "An approach to trust case development", *Proc. of 22nd International Conference on Computer Safety, Reliability and Security (SAFECOMP 2003)*, LNCS Vol. 2788, pp. 193-206, (2003).
[13] J. Górski, A. Jarzębowicz, J. Miler, M. Witkowicz, J. Czyżnikiewicz, P. Jar, "Supporting Assurance by Evidence-Based Argument Services", *Proc. of SAFECOMP 2012*, LNCS 7613, pp. 417-426, (2012).
[14] J. Górski, A. Jarzębowicz, J. Miler, "Comparative conformance cases for monitoring multiple implementations of critical requirements", *Proc. of 32nd International Conference on Computer Safety, Reliability and Security (SAFECOMP 2013)*, LNCS Vol. 8153, pp. 8-20, (2013).
[15] W. S. Greenwell, J. C. Knight, C. M. Holloway, J. J. Pease, "A taxonomy of fallacies in system safety arguments", *Proc. of the 2006 International System Safety Conference*, (2006).
[16] S. Grigorova, T. Maibaum, "Argument evaluation in the context of assurance case context modeling", *Proc. of IEEE Int. Symposium on Software Reliability Engineering Workshops*, (2014).
[17] GSN Community Standard Working Group, "GSN community standard version 1", (2011).
[18] R. Hawkins, T. Kelly, "A structured approach to selecting and justifying software safety evidence", *Proc. of 5th IET International System Safety Conference*, Manchester, UK, IET, (2010).
[19] R. Hawkins, T. Kelly, J. Knight, P. Graydon, "A New Approach to creating Clear Safety Arguments", *Proc. of 19th Safety Critical Systems Symposium*, (2011).
[20] ISO/IEC, "ISO/IEC 15026-2:2011 Systems and software engineering – Systems and software assurance – Part 2: Assurance case", (2011).
[21] T. Kelly, "Reviewing Assurance Arguments - A Step-by-Step Approach", *Proc. of Workshop on Assurance Cases for Security*, Edinburgh, UK, (2007).
[22] Z. Langari, T. Maibaum, "Safety cases: a review of challenges", *International Workshop on Assurance Cases for Software-intensive Systems (ASSURE 2013)*, San Francisco, (2013).
[23] N. Leveson, "The use of safety cases in certification and regulation", *Journal of System Safety*, Vol. 47, No. 6, System Safety Society, (2011).
[24] Object Management Group, "Structured Assurance Case Metamodel (SACM)," version 1.0, (2013).
[25] B. R. Larson, "Open PCA Pump Assurance Case", SAnToS research group, Kansas State University, http://openpcapump.santoslab.org/, (2014).
[26] G. Shafer, "Mathematical theory of evidence", Princetown University Press, (1976).
[27] S. Toulmin, "The Uses of Argument", Updated Edition, Cambridge University Press, (2003).
[28] B. Verheij, "Evaluating arguments based on Toulmin's scheme", *Argumentation* 19 (3), pp. 347-371, (2005).
[29] D. Walton, "Defeasible reasoning and informal fallacies", Synthese Vol. 179, no. 3, pp. 377-407 (2011).
[30] R. Weaver, P. Mayo, T. Kelly, "Gaining Confidence in Goal-based Safety Cases", *Proc. of 14th Safety Critical Systems Symposium*, Springer, (2006).

	Defeater description	**% found**
I-1	No defined inference rule. It is unknown how the conclusion is drawn from premises.	3,1%
I-2	Wrong inference rule	9,4%
I-3	Inference rule not sufficiently justified (unknown reasons, doubts about completeness)	9,4%
I-4	Inference rule incomplete: not all requirements regarding premises specified. As a result, some premises may be missing.	17,3%
I-5	Inference rule leaves out a specific situation/factor or counterevidence. As a result, some premises are missing.	5,5%
I-6	Inference rule does not consider mutual influence of premises.	4,7%
P-1	One or more premises required by the inference rule are missing	11,0%
P-2	Superfluous premise(s) of unclear role in the inference rule.	0,8%
P-3	Superfluous premise(s) which should be a part of the confidence argument instead of the assurance argument.	0,8%
P-4	Unreliable or faulty premise(s).	15,7%
P-5	Premise not properly defined (unclear, unverifiable).	15,7%
C-1	Required context not specified.	4,7%
C-2	Wrong context assumed.	1,6%

Table 1. Defeater checklist