An Approach to Evaluation of Arguments in Trust Cases

Łukasz Cyra and Janusz Górski
Gdańsk University of Technology, Narutowicza 11/12, 80-952 Gdańsk, Poland
lukasz.cyra@eti.pg.gda.pl, jango@pg.gda.pl

Abstract

Trustworthiness of IT systems can be justified using the concept of a trust case. A trust case is an argument structure which encompasses justification and evidence supporting claimed properties of a system. It represents explicitly an expert's way of assessing that a certain object has certain properties. Trust cases can be developed collaboratively on the basis of evidence and justification of varying quality. They can be complex structures impossible to comprehend fully by a non-expert. A postulated model of communicating trust case contents to an ‘ordinary’ user is an expert acting on user’s behalf and communicating his/her assessment to the user. Therefore, a mechanism for issuing and aggregating experts’ assessments is required. The paper proposes such a mechanism which enables assessors to appraise strength of arguments included in a trust case. The mechanism uses Dempster-Shafer’s model of beliefs to deal with uncertainty resulting from the lack of knowledge of the expert. Different types of argumentation strategies were identified and for each of them appropriate combination rules were presented.

1. Introduction

Potential users of a system need to be convinced to use it, owners of a system are interested in different ways of convincing the users, assessors are expected to assess a system against a given set of criteria and developers need support to develop a trustworthy system. Therefore, trustworthiness of IT systems is a crucial issue and appropriate methods of tackling this problem are required.

Trust-IT framework [1-3] aims at development of trust cases which are (electronic) documents that provide satisfactory (from a given viewpoint) justification for a specific set of claims (regarding system’s properties considered for a given application in a given environment) to make a judgement about their trustworthiness. The underlying idea is to make expert judgement explicit in order to redirect dependence on judgement to issues on which we can trust this judgement [4].
In development of a complex system the complexity of the corresponding trust case grows and it may easily become too difficult for a reader to analyse the whole of it. Moreover, as trust cases often include interdisciplinary knowledge, their thorough assessment may require involvement of several experts. Therefore, a mechanism that supports aggregation of partial assessments and presentation of the final judgement in a simple and understandable way is required.

In this article a method of evaluation of the strength of the argumentation kept in a trust case and a mechanism of issuing recommendations are presented. It is shown how partial assessments can be aggregated and how it leads to the final assessment of the topmost claim of the trust case. Different types of arguments are identified and it is shown how they influence the judgement of their conclusions, by proposing for each argument type the corresponding aggregation equation.

The appraisal mechanism is a continuation of the research presented in [5]. Formerly, a different method of trust case appraisal was proposed, which unfortunately was not in line with adopted practices of trust case development. Therefore, some basic assumptions concerning the mechanism had to be changed. This resulted in a complete rework of the method among which the mechanism of issuing expert opinions was changed, existence of different types of arguments was taken into account and for each of them an aggregation rule was proposed.

2. Trust case

A trust case is an argument structure that comprises evidence and justification for a certain claim. A trust case has a tree-like form and is composed of nodes of different types. Each node contains a statement which constitutes a part of the argument, and the type of the node signifies the role of the statement in the argumentation. The argument structure is represented in a graphical form.

The basic ‘building block’ consists of a conclusion to be justified (a claim), the premises which are referred to in order to justify the conclusion and the justification which explains why the conclusion results from its premises. Such a block is called argument and is shown in fig. 1. This model follows the Toulmin model of argument [6] and can be applied to represent formal inference as well as informal argumentation.

The claims in the premises can be further demonstrated by providing more detailed arguments and this way the argument structure can grow recursively until the premises do not contain further (sub-)claims.

In fig. 1, an arrow represents that a node of a given type can be a child of the node pointed to by the arrow.
The statement to be justified (the conclusion) is represented as a claim (denoted $\text{Claim}$). The argument strategy (denoted $\text{Argument}$) contains the basic idea of how to derive a conclusion from premises. In case of counter-arguments (denoted $\text{Counter-Argument}$) it includes the idea of rebuttal of the claim. The justification of the inference from the premises to the conclusion is represented as a node of type warrant (denoted $\text{Warrant}$). A premise can be of three different types: it can represent an assumption (denoted $\text{Assumption}$), in which case the premise is accepted without further justification; it can be a (more specific) claim which is justified further; or it can represent a fact (denoted $\text{Fact}$) which is obviously true or otherwise is supported by some evidence. The evidence is provided in external (to the trust case) documents which are pointed to by nodes of type reference (denoted $\text{Reference}$).

Everywhere in the argument tree an information node (denoted $\text{Information}$) can be placed and any node can be linked to the information node. Such nodes contain explanatory information which does not constitute part of an argument. In addition, links of appropriate types (denoted $\text{Link}$) can be used to point to a specific part of a trust case (if this part is to be re-used).

3. Appraisal of the trust case

Two objectives of the appraisal mechanism are to provide means of the assessment of the compelling power of the trust case and to make it possible to communicate the results to non-experts. The objectives are addressed by the following scenario:

(1) A trust case is made accessible to a group of experts called assessors. Assessors can express their opinions about the ‘quality’ of the evidence and about the validity of the arguments included in the trust case. They issue assessments only about the simple elements of the structure (i.e. assumptions, facts and warrants, which are not supported by an argument).

(2) The opinions are then aggregated into assessment of the claims. It is applied recursively starting from the leaves of the argument tree and propagates the assessments to the topmost claim. The aggregation results must be consistent with what would be deduced by an expert in such a situation.
To calculate assessments, the appraisal mechanism adopts the model shown in Fig. 2. It provides for expressing assessments in the discrete Assessment scale.

**Assurance scale**

- for sure
- with very high assurance
- with high assurance
- with low assurance
- with very low assurance

**Truthfulness scale**

- true
- partially true
- partially false
- false

**Assessment scale**

- for sure true
- with very high assurance true
- with high assurance true
- with low assurance true
- with very low assurance true
- for sure partially true
- partially true
- partially false
- false

**Figure 2. Trust case assessment scale**

The Assessment scale is a product of two more specific scales, the Assurance scale which distinguishes five assurance levels (‘for sure’, ‘with very high assurance’, ‘with high assurance’, ‘with low assurance’, ‘with very low assurance’) and the Truthfulness scale with four truthfulness levels (‘true’, ‘partially true’, ‘partially false’, ‘false’). Additionally, one level representing a complete lack of knowledge was added (‘with no assurance true’).

The assessment of a premise proceeds as follows. If an assessor does not have any reasons to accept or reject the premise, she/he issues a recommendation ‘with no assurance true’. Otherwise, the truthfulness of the premise is assessed using the Truthfulness scale, and the available evidence (or common knowledge) that supports the truthfulness judgement is assessed using the Assurance scale. If these two assessments are combining, a final recommendation from the Assessment scale is obtained.

For instance, if a given premise receives with high assurance on the Assessment scale and true on the Truthfulness scale, then the result on the Assessment scale is with high assurance true.

The assessment of the inference (warrant) proceeds in a similar way. If an assessor does not see reasons to accept or reject the warrant, she/he issues a recommendation ‘with no assurance true’. Otherwise, the truthfulness of the warrant is assessed by assessing to which extent the truthfulness of the premises influences the truthfulness of the conclusion. If the premises are irrelevant to the conclusion, then obviously the assessment is ‘false’. For inductive arguments (reasoning from examples to generalization) the assessment is likely to be ‘partially true’ or ‘partially false’, depending on the specific case whereas deductive arguments are likely to be assessed as ‘true’. Then it is assessed to which extent available evidence (or common knowledge) supports the truthfulness of the inference. By combining these two assessments the final recommendation is obtained.

Assessments are represented in a numerical form. Two representations are used:

1. \( <As(i), Tr(i)> \in [0,1] \times [0,1] \)
2. \( i \) – assessed item,
As(i) – numerical representation of assurance estimation, where As(i) = 0 represents assessment 'with no assurance true', and As(i) = 1 represents assessment 'for sure'.

Tr(i) - numerical representation of truthfulness estimation, where Tr(i) = 0 represents assessment 'false', and Tr(i) = 1 represents assessment 'true'.

(2) \[<Bel(i), Pl(i)> \subseteq [0,1] \times [0,1]\]

i – assessed information,

Bel(i) – Dempster-Shafer’s belief function [7], representing the amount of belief that directly supports a given information.

Pl(i) – Dempster-Shafer’s plausibility function [7], representing the upper bound on the belief that can be gained by adding new evidence (because there is so much evidence that contradicts it).

Both representations are equivalent, which is shown by the following bijection functions.

\[
\begin{align*}
Bel(i) &= Tr(i) \cdot As(i) \\
Pl(i) &= 1 - As(i) \cdot [1 - Tr(i)] \\
As(i) &= Bel(i) + 1 - Pl(i) \\
Tr(i) &= \begin{cases} 
Bel(i)/[Bel(i) + 1 - Pl(i)] & Bel(i) + 1 - Pl(i) \neq 0 \\
1 & Bel(i) + 1 - Pl(i) = 0
\end{cases}
\end{align*}
\]

i – assessed information

4. Types of inference

Aggregation of partial assessments to the assessment of the conclusion of an argument is not a simple issue, the basic difficulty is that it has to reflect the (possibly different) ways the premises influence the conclusion.

In trust cases there are two situations in which partial assessments are aggregated:

1) Calculation of the assessment of a claim which is supported by more than one argument (and/or counter-argument).

2) Calculation of the assessment of a claim which is the conclusion of a single argument taking into account the assessments of the premises and the warrant of this argument.

The inference types frequently occurring in trust cases can be divided into two main categories:

Type 1: Arguments for which the set of premises constitutes the sufficient set of conditions for accepting the conclusion, and falsifying a single premise leads to the rebuttal of the conclusion, or it leads to a situation in which the argument does not provide any information for and against the conclusion.

Type 2: Arguments in which a rebuttal of one of the premises not necessarily leads to the rejection of the conclusion or lack of information from the inference. In such cases the belief in the conclusion decreases, but if the remaining premises are accepted, the conclusion can still be attained.

An actual argument that does not comply with either Type 1 or Type 2 can usually be represented as a combination of the Type 1 and Type 2 arguments.
Arguments of Type 1 can be further divided into 2 sub-types.

1.1) Each of the premises is a necessary condition for the conclusion. In this case it cannot be replaced by any other premise. This type of argument is called *Necessary and sufficient condition list* argument.

1.2) Neither of the premises is a necessary condition for the conclusion and each can be possibly replaced by another one while still preserving the Type 1 of the argument. This type is called *Sufficient condition list* argument.

If an argument of Type 1 does not comply with the 1.1 or 1.2 subtype, it can be represented as a combination of arguments of subtype 1.1 or 1.2.

Arguments of Type 2 can be further divided into 3 sub-types.

2.1) Each of the premises supports a different part of the conclusion.

Let us consider as an example an argument in which trustworthiness of a system is argued by demonstrating safety, security and privacy (assuming that safety, security and privacy are independent qualities of the system).

Such a situation can be represented graphically as shown in fig. 3 (case 2.1). Each region represents a part of the conclusion, and overlapping of the regions (if any) would indicate covering the same aspect.

This type of argument is called *Complementary arguments.*

![Figure 3. Argument types](image)

2.2) Each of the premises supports the whole conclusion in a way independent from other premises. This type of argument is called *Alternative arguments.*

Let us consider an example in which we argue system reliability by referring to the results of system tests and in addition to the results of code inspections (see fig. 3, case 2.2). Note however, that this argument type can be split into two arguments, each supporting the claim of reliability by referring to only one premise.

2.3) Each of the premises supports a (not necessarily disjoint) part of the conclusion.

Let us consider an example procedure which takes $x$ as an input and returns $[f_1(x), f_2(f_1(x))]$ as its output. We can claim that an implementation of the procedure is correct referring to the facts that both parts of the output are correct. However, the correctness of the first part of the output supports the conclusion only partially, whereas the correctness of the second part of the output supports the whole conclusion (even if we do not know if the first part is correct).

In theory, this case could be treated as a combination of subtypes 2.1 and 2.2 providing that we would be able to identify precisely the overlapping part of the
conclusion and to split it in such a way that the argument of subtype 2.3 were represented as a combination of arguments of subtypes 2.1 and 2.2. In practice however, it is not always obvious where the overlapping part is and an individual treatment of each argument of subtype 2.3 would need concerning the aggregation rules for the assessments.

To cope with the above problem, we have adopted a pragmatic solution in which, if overlapping is insignificant, arguments of subtype 2.3 are treated as arguments of subtype 2.1 (complementary arguments) and, if overlapping is significant, they are treated as arguments of subtype 2.2 (alternative arguments). If such simplification is invalid, the aggregation rule for the argument has to be considered individually. In both cases the decision is left to the argument proposer.

5. Aggregation equations

Different types of arguments described in the previous section were analysed with the aim of capturing the way of expert thinking while assessing such arguments. As a result, aggregation equations were defined for each type of an argument.

In the following equations it is assumed that: $s$ is a claim; $a_i$ is the $i^{th}$ premise or argument; $w$ is a warrant; $k_i$ is a weight of $i^{th}$ premise defined by the argument proposer.

5.1. Alternative arguments

For alternative arguments Yager’s modification of Dempster’s rule of combination [8] was proposed (in the version for two alternatives):

$$Bel(s) = Bel(a_1) \cdot Bel(a_2) + Bel(a_1) \cdot [Pl(a_2) - Bel(a_2)] + Bel(a_2) \cdot [Pl(a_1) - Bel(a_1)]$$

$$Pl(s) = 1 - [1 - Pl(a_1)] \cdot [1 - Pl(a_2)] - [1 - Pl(a_1)] \cdot [Pl(a_2) - Bel(a_2)] - [1 - Pl(a_2)] \cdot [Pl(a_1) - Bel(a_1)]$$

When more than two arguments exist, appropriate modifications of these equations should be applied.

Recommendations coming from counter-arguments should be transformed before applying the rule of combination in accordance with the following equations:

$$Bel_{\text{argument}}(a_i) = 1 - Pl_{\text{counter-argument}}(a_i)$$

$$Pl_{\text{argument}}(a_i) = 1 - Bel_{\text{counter-argument}}(a_i)$$

Example: Let us assume that in the example 2.2 in fig. 3 the code inspection and the tests demonstrated reliability with high assurance, i.e. they are assessed as ‘with high assurance true’. Using the presented rule of aggregation assessment ‘with very high assurance true’ is obtained. This example shows that by aggregation of assessments coming from two independent arguments the level of confidence is heighten, which is in accordance with the intuition.

5.2. Necessary and sufficient condition list argument

For the necessary and sufficient condition list argument type the following equations were proposed:

$$Bel(c) = Bel(w) \cdot Bel(a_1) \cdot Bel(a_2) \cdots Bel(a_n)$$

$$Pl(c) = 1 - Bel(w) \cdot [1 - Pl(a_1) \cdot Pl(a_2) \cdots Pl(a_n)]$$

Example: Let us assume that we infer a conclusion on the basis of two premises. The warrant is assessed as ‘for sure true’ and both of the premises are assessed as ‘with high assurance partially true’. As a result of aggregation ‘with low assurance partially true’ is obtained. So it
can be noticed that, as it could be expected, the fact that the premises are uncertain caused that the conclusion is less certain than each of them. The truthfulness level remained the same as both of the premises contain the same ratio of positive and negative evidence.

5.3. Sufficient condition list argument

For the sufficient condition list argument type the following equations were proposed:

\[ \text{Bel}(s) = \text{Bel}(w) \cdot \text{Bel}(a_1) \cdot \text{Bel}(a_2) \cdot ... \cdot \text{Bel}(a_n) \]

\[ \text{Pl}(s) = 1 \]

Example: Let us consider a similar example as above. In the case of this type of inference assessment ‘with low assurance true’ is obtained. This is caused by the fact that from this type of inference we cannot build disbelief in the conclusion. On the other hand, because the premises are not certain, the trust in the conclusion is even lower than in each of them.

5.4. Complementary arguments

For complementary arguments the following equations were proposed:

\[ \text{Bel}(s) = \text{Bel}(w) \cdot \frac{k_1 \text{Bel}(a_1) + k_2 \text{Bel}(a_2) + ... + k_n \text{Bel}(a_n)}{k_1 + k_2 + ... + k_n} \]

\[ \text{Pl}(s) = 1 - \text{Bel}(w) \cdot \left(1 - \frac{k_1 \text{Pl}(a_1) + k_2 \text{Pl}(a_2) + ... + k_n \text{Pl}(a_n)}{k_1 + k_2 + ... + k_n}\right) \]

Example: Let us assume that in the example 2.1 in fig. 3 the warrant is assessed as ‘with high assurance true’, safety has priority 3 and is assessed as ‘with very high assurance true’, security has priority 2 and is assessed as ‘with low assurance partially false’ and privacy has priority 2 and is assessed as ‘with low assurance partially false’. By application of the above rule assessment ‘with low assurance partially true’ is obtained. So it can be noticed that the result of the aggregation of the assessments is a kind of weighed mean value diminished a bit by the fact that the warrant is not fully trustworthy.

6. Conclusions

In this article we proposed a method of evaluation of arguments that occur in trust cases. The method is to be used as a tool supporting assessors of trust cases. With this method, an assessor evaluates the premises (facts and assumptions) of the lowest level arguments and the warrants of all arguments maintained in the trust case. Then, the evaluation of the conclusions is performed automatically by applying the adequate aggregation equations.

The proposed solution has been already validated in some experiments where the assessments produced by application of aggregation equations were compared with expert judgements. The experiments demonstrated that the application of the method delivers results that are close to expert evaluations while significantly reducing the load put on the expert.

The presented method is a subject of intensive development. It is going to be implemented and used in evaluation of real trust cases developed for e-Health services. It will also be further validated in a series of experiments involving larger groups of
experts in order to assess how closely the proposed aggregation equations reflect the way of thinking of an ‘average’ expert and to modify them, if necessary.

7. References


