ARGEVIDE

TRUST-IT Guide

Part 1. Structure of an Assurance Case

v1.6, 22.08.2022

1 Summary 2
2 Introduction to TRUST-IT method 3
3 Assurance case models and notations 4
4 Claims 5
5 The concept of an argument 7
6 Reasoning steps in the argument 8
7 Evidential steps in the argument 11
8 Evidence 13
9 Assumptions 14
10 The complete argument 15
11 Argument structure definition 17
12 How TRUST-IT relates to other assurance case approaches? 20

1 Summary

TRUST-IT is a method of collaborative and iterative assurance case deployment which aims at development of strong and defensible arguments. The method is based on the argument model described in this document and its extensions. The extensions enable the development of an argument in a way that enables continuous verification and feedback in the form of the assessment and also provide support for change management and versioning. Basic information about the method is given in Section 2.

Argument models can be presented in several ways. TRUST-IT arguments are presented using hierarchical notation and graphical GSN notation. These two ways of argument presentation are equivalent and they can be used interchangeably. This is presented in Section 3.

TRUST-IT method is focused on the process of argument development, verification and assessment. It does not define a new graphical notation but introduces some specific requirements on the argument information model to strengthen the argument. Elements of the argument structure, which are presented in sections 4 and following, include:

- the concept of a claim which is the main building element of the argument,
- reasoning steps in the argument structure,
- evidential steps to connect the argument with the supporting evidence,
- references of evidence,
- assumptions.

The complete definition of TRUST-IT argument structure is presented in Section 11.

The main differences between TRUST-IT argument model and the argument structure defined by GSN Community Standard are described in Section 12.

TRUST-IT method is implemented by NOR-STA web application which also implements assurance case models defined in GSN Community Standard and ISO 15026 standard. Please check other Argevide publications for more information about NOR-STA tool and explanations how TRUST-IT relates to ISO 15026 standard, GSN Community Standard and OMG Structured Assurance Case Metamodel (SACM).

2 Introduction to TRUST-IT method

An **assurance case** is a structured, compelling argument, supported by evidence, justifying that a system has some postulated properties in a specific context and environment. The postulated properties are usually system safety, security, availability or compliance with relevant standards. Assurance case focused on one these properties can be named according to it. The name “safety assurance case” can be used for an argument related to system safety.

The main components of TRUST-IT assurance case approach are presented in Figure 1.

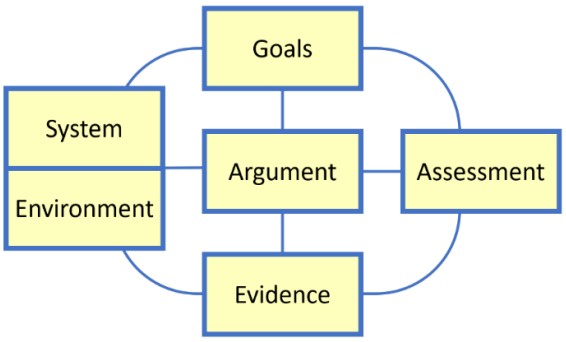


Figure 1. The main components of TRUST-IT assurance case

Goals specify properties of the system that are to be demonstrated by the assurance case. They have to be specified to start development of an assurance case.

The goals are specified in the context of the **system** and its **environment**. The system is the subject of assurance goals. It can be a simple device, a complex system or a set of integrated systems, an organization or a process. The system operates in the environment which covers operational physical objects, regulations and any information external to the system. Description of the system and the environment should be sufficiently detailed to allow for unambiguous interpretation of the goals, the argument and evidence.

The **argument** contains an explicit and verifiable reasoning supported by evidence which demonstrates that the specified goals are achieved. It is the core element of an assurance case.

The argument is to be supported by **evidence**. Evidence is a verifiable and auditable information in any form (documents, data, photos, video, statistics) which describes the system, its components, characteristics, properties or events including data on the system operation. To be valid, the evidence needs to be consistent with the reality and up to date.

The last component of an assurance case is the **assessment**. The assessment is produced as a result of a systematic review of the argument and evidence. It gives information if the argument and evidence support the claims that the goals are achieved and that the system has required properties.

In this part of TRUST-IT Guide we present the structure of an assurance case.

3 Assurance case models and notations

Before we start presenting TRUST-IT assurance case method we have to distinguish between assurance case information model and notation. The information **model** defines the way how assurance case data is structured and stored, processed and transferred by computers. The data may be stored using XML, JSON or a database data format. The model defines the technical representation of an assurance case.

The **notation** is the way how we organize information to present it to the user. Assurance cases can be presented in a number of ways. The argument structure can be presented using graphical notations like GSN or CAE, textual notation or hierarchical notation, or in a tabular format. A view defined by a notation does not have to contain the complete set of information included in the assurance case. Some notations can be applied to specific information models, e.g. the argument structures of GSN and CAE differ, but some notations like a tabular format can be applied to any model of assurance case information.

TRUST-IT method uses two notations to present the argument structure for users. A hierarchical notation is a default representation to develop and manage the argument. You can also present TRUST-IT arguments using the graphical Goal Structuring Notation (GSN). All assurance case views in hierarchical and graphical notation presented in this document have been generated using NOR-STA.

A comparison of hierarchical and GSN notations is presented in Figure 2. The hierarchical notation is shown on the left. The notation uses indentation to represent hierarchy of the argument. The plus and minus icons (and) are used in the tool to expand and collapse the argument elements below. This notation is useful for editing the argument structure as users are less distracted with arranging the layout of the diagram.

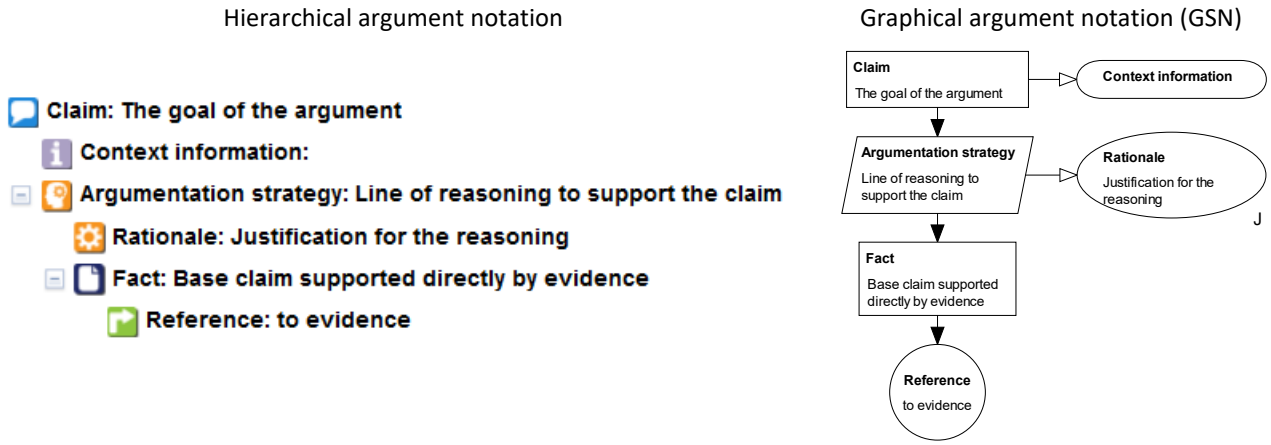


Figure 2. An argument in NOR-STA hierarchical notation (left) and a corresponding GSN diagram (right)

Different arrow shapes are used in GSN diagrams to present relations “is supported by” (black arrows) and “in the context of” (hollow arrows). Types of relations are not presented in the hierarchical notation but they can be determined depending on the type of connected elements.

4 Claims

An assurance case is developed for a **goal** which defines the required properties of a system like safety, security, availability or other. Goals are represented by claims and their context information.

A **claim** is a true-false statement, a predicate which defines an object, its properties and conditions.

A sentence “Device is adequately safe in operation” is a true-false statement as it can be true or it can be false. A claim should be defined in a precise and unambiguous way to enable the judgement if it is true or not.

In general a claim follows a schema: “an object has specified properties in specified conditions and context”. For the claim presented above we can say that:

- an object is a “device”
- the property is “adequate safety” and
- the conditions is the “operation of the device”.

These should be defined in a precise way. For example we should know answers for questions:

- What device are we talking about? Can we specify its type? Is it a series of devices or one unit?
- How “safety” is defined and how it can be measured in an objective way?
- What are the criteria for safety to be “adequate”? How this can be judged or measured?
- What are the conditions of the “operation” when safety is required? When the “operation” occurs?

Description of a claim can contain all information to answer the questions and provide precise definition. Alternatively, the claim can be accompanied by context elements which provide the needed information. Usually the relevant information is distributed in different documents and it is practical to add context information elements to a claim as presented in Figure 3 (we use GSN graphical notation to present the argument in this document). The rectangle C1 represents a claim and the other shapes with rounded sides represent context information.

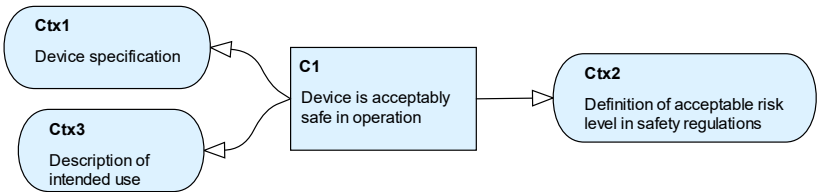


Figure 3. Top claim of the argument (GSN diagram)

When you analyse the meaning of a claim you should always take into account the attached context information. The context information is also applicable for all argument elements supporting a given claim. This is called an inherited context. According to this rule, when any argument element refers to a “device” it should be interpreted in line with Ctx1 description unless explicitly specified otherwise.

We use diagrams like Figure 3 to present the structure of an argument, but not all argument information is presented on diagrams. Only element labels (e.g. C1 or Ctx1) and names (e.g. “Device is adequately safe in operation” or “Device specification”) are presented on diagrams. What is also a part of an argument element but not presented on a diagram is the element description. Descriptions are managed in assurance case editor and are also presented in assurance case reports.

When you develop an argument you may decide when you want to create dedicated context elements and when you want to put the information into a claim description. The claim definition presented in Table 1 is equivalent to the diagram in Figure 3.

Table 1. A variant of claim C1 definition

Label	C1
Name	Device is adequately safe in operation
Description	Device specified in “Device Specification” (https://repository/doc/Dev/DevSpec.pdf) satisfies safety requirements defined in the standard ISO/IEC 12345 (https://repository/doc/std/iso-12345.pdf) when operated according to “Device Operation Manual” (https://repository/doc/Dev/OperationManual.pdf)

The objective of a diagram is to help the users understand the argument and to navigate through it. The diagram is not to provide the full detailed information. When you review an argument you have to browse descriptions of all argument elements and also supporting evidence (which will be described in a following section in this document).

The claim is a statement that should be defined in a clear, explicit and understandable form and it should define:

- a referred object which can be a physical or logical entity, for example a software function,
- the required properties of the object,
- the context, conditions and limitations when the property is required:
- environmental conditions (temperature, humidity, vibrations, pressure, radiation and others),
- range or extreme values of size, volume, load, speed or other attributes,
- conditions on input and processed data,
- time limitations and dependencies,
- system or environment architecture, components, relations or boundaries,
- human or technology limitations,
- and other applicable conditions and limitations;
- acceptance criteria when the property should be recognized to be satisfied,
- for base claims, which are supported directly by evidence, the acceptance criteria should be sufficiently detailed to guide the evidence review process and verify if it fully supports a given claim.

Claims are the core elements of each assurance case and the argument quality greatly depends on how precise we define claims. Quite often weak assurance case starts with poorly defined claims and their context.

5 The concept of an argument

Development of an arguments starts when the claim which specifies the goal is defined. You may define more then one claim when necessary depending on the specified goals. Such claims are called **top level claims**. A separate argument is to be developed for each top claim.

The argument justifies how the evidence supports the goals specified for an assurance case. The argument should be convincing and verifiable to get the acceptance and approval which is usually required to proceed with a system deployment.

The argument is developed in a systematic way and is divided into steps. There are two types of argument steps:

- **reasoning steps** when claims are supported by other claims and
- **evidential steps** where claims are supported directly by evidence (such claims are called "base claims").

A **reasoning step** describes the reasoning how a given claim can be inferred from its supporting claims. When we develop an argument for the device safety we can say the device "is adequately safe" because all safety functions are implemented. In the next step we demonstrate that all safety functions have been implemented. Our strategy of reasoning can be based on the use of a systematic development process. The reasoning steps are denoted with arrows in Figure 4. Indents indicate the hierarchy of argument elements. The direction of the reasoning is from the evidence to the claims as indicated by the direction of arrows.

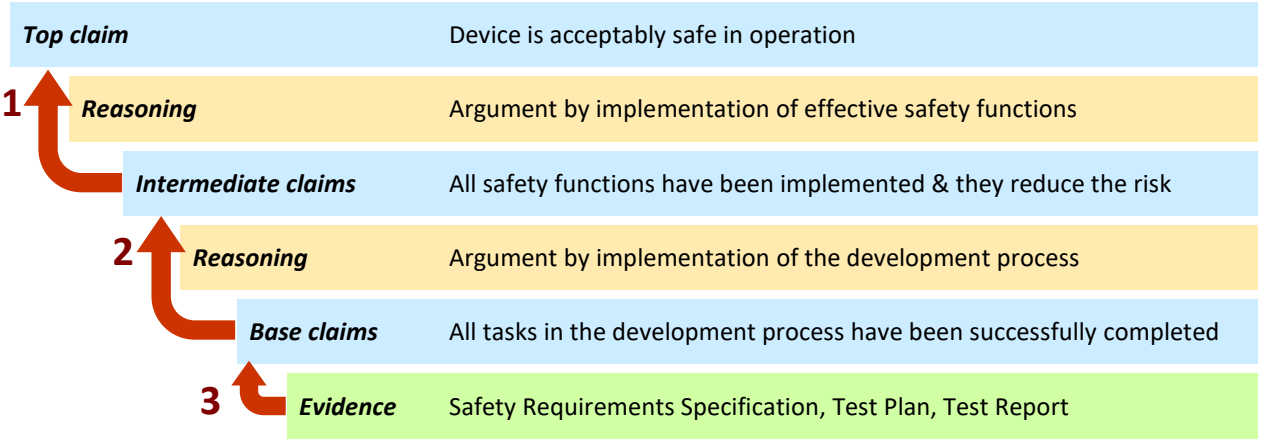


Figure 4. General argument structure and the flow of reasoning

The whole reasoning is based on the evidence. The lowest level of the argument consists of a set of **evidential steps**. The objective of this step is to establish claims backed directly by evidence. In our case study of a device safety argument, the documents like Test Plan and Test Report can be used to demonstrate that the required activities of the development process had been completed.

The claims supported directly by evidence are called **base claims** or **facts**. A fact is a statement (a claim) based directly on observation of real-world artefacts (the evidence). When I look at a thermometer and I see the temperature outside is 18 degrees Celsius then based on this evidence I can state a fact "the temperature outside is 18°C". The statement of a fact usually requires some interpretation of the available evidence, but not reasoning. For each fact a systematic way of determining if it is true or not for a given available evidence should be established.

It is important to distinguish base claims as they should provide precise acceptance criteria when an evidence item can be accepted to correctly support it.

6 Reasoning steps in the argument

Reasoning steps describe how some claims are supported by other claims. The argument starts with a top claim supported by intermediate claims and they, in turn, are supported by other claims until we get to the level of base claims supported directly by evidence.

For each reasoning step we should specify:

- a **strategy** which describes the way of reasoning,
- a **rationale** which justifies that the strategy is applicable and correctly applied,
- the supporting **premises** (claims),
- and optionally **assumptions** if the reasoning requires them.

The argument usually is developed top-down and we will present the reasoning steps in this direction. The reasoning for the top claim in based on the rule that safety functions may reduce the risk level of the system to an acceptable level.

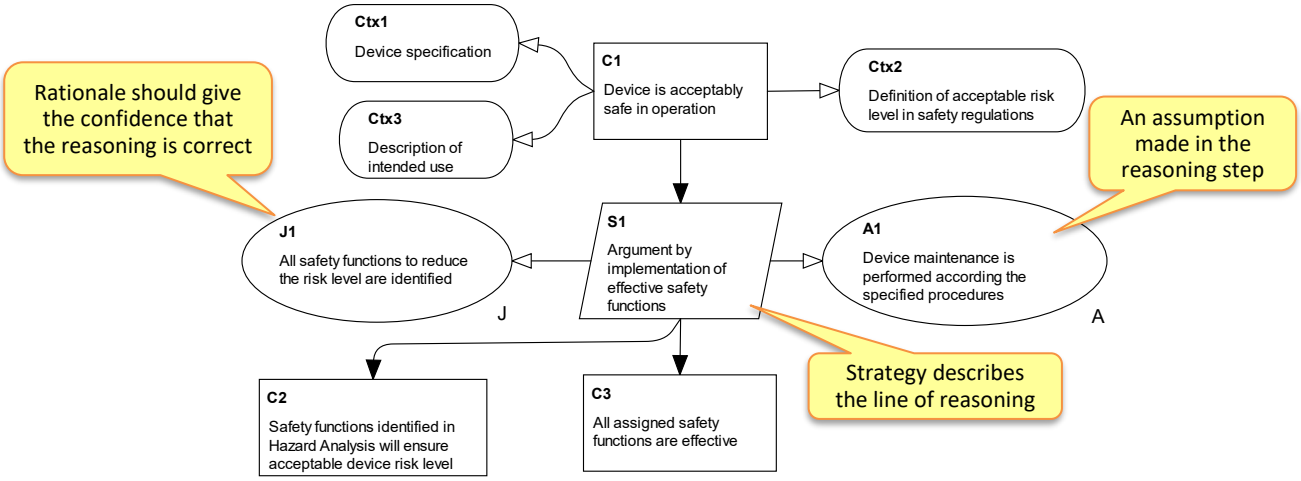


Figure 5. Reasoning step for the top claim (GSN diagram)

A **strategy** specifies the inference rule used to reason that the conclusion (the higher-level claim) is true when the premises (the lower-level claims) are true.

A strategy is not a statement and it is not a predicate. It just identifies a rule we use to reason. You can say a strategy is like a recipe telling you how to take ingredients and combine them to bake a cake. Or like a formula to calculate a result from the available data.

The strategy should identify the required premises and describe the inference rule between them and the supported claim which is the conclusion of the reasoning.

A **rationale** is a statement which justifies that the right strategy is used to support a given claim and that it is used in the right way. The rationale says we know HOW to check and demonstrate that the goal is achieved. When the rationale is wrong we are not able to say if the goal like safety is achieved. It should explain why we think our approach of reasoning or measurement is valid.

The objective of explicit definition of a rationale is to build the **confidence** in the reasoning. It will say if we can trust the reasoning leading to the conclusion about the top claim of the argument. We will discuss this in detail in the Part Two of this Guide.

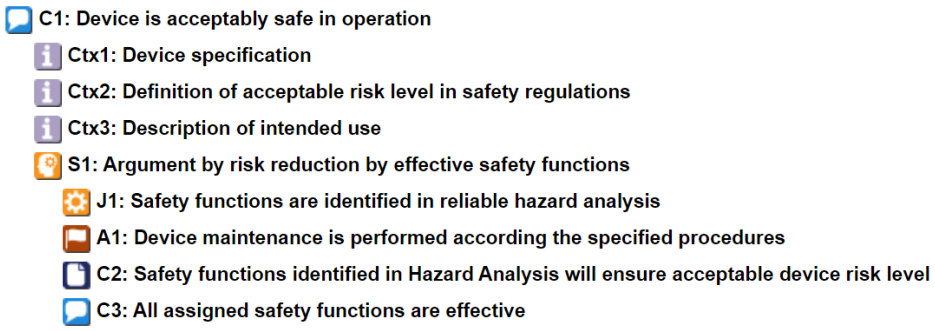


Figure 6. The reasoning step for the top claim (NOR-STA notation)

The reasoning to support the top claim is based on the rule that effective safety functions reduce the risk of specific hazards. When we implement a sufficient set of safety functions then the overall risk for the device will be reduced to a level that can be accepted for safe operation according to the regulations.

There is an additional condition for this rule, namely it is required that the device is properly maintained during the operation. This condition is defined as an assumption. We will discuss assumptions in a further section in this Guide.

The strategy says we use effective safety function to reduce risk to an acceptable level. We say a safety function is “effective” when it reduces the risk according to the hazard analysis.

To provide confidence that this strategy is efficient we should make certain the hazard analysis gives us the right list of safety functions to be implemented and verified if they are effective. Therefore we define a rationale with states that the hazard analysis is reliable in providing safety functions to reduce the risk level.

To summarize the roles of strategies and rationales are:

- **strategy** explains what we do, how we reason to achieve the goal,
- **rationale** (justification) says when we can have confidence it works correctly.

The topic of confidence will be discussed in Part 2 of this Guide.

So far we have discussed the first reasoning step (strategy S1) which supports the top claim. It is supported by two claims: C2 and C3. C2 is a base claim (fact) supported directly by evidence and we will discuss it in the next section. Now we will focus on the claim C3 “All assigned safety functions are effective”. To support this claim we need a way how to demonstrate that the safety functions are effective.

To keep our case study small, we assume we have a reliable development process that guarantees effectiveness of safety functions when all safety requirements are verified. The main milestones of the process are:

- specification of safety requirements for safety functions,
- verification and validation planning and
- independent verification and validation.

Based on the development process we can formulate a strategy that effectiveness of safety functions is achieved by implementation of the development process. The required premises for the argumentation strategy would be related to the three mandatory milestones of the process. This reasoning step is presented in Figure 7 (GSN diagram) and in Figure 8 (NOR-STA notation).

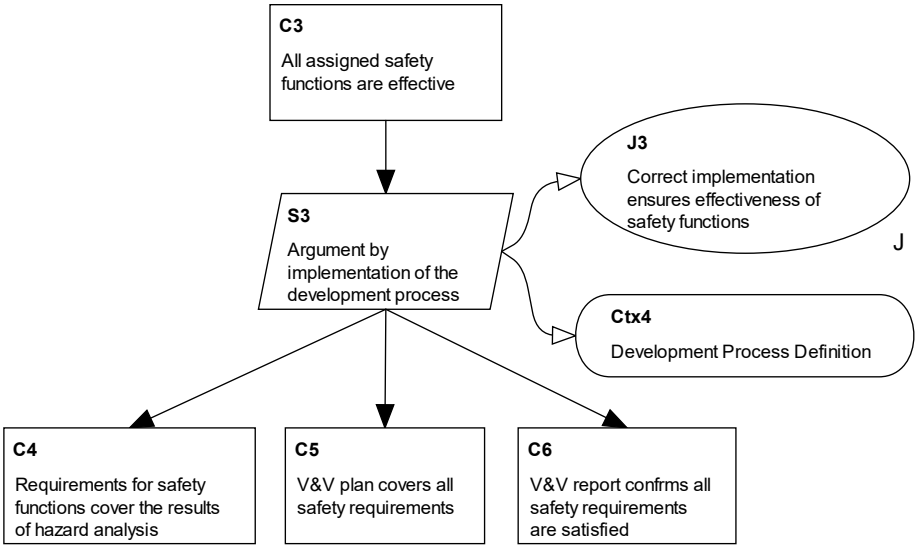


Figure 7. Reasoning step for claim C3 (GSN diagram)

The reasoning step refers to the Development Process and we add context information with the definition of the process. This information allows the reviewers to check if the strategy correctly follows the process, if all the requirements of the process are satisfied (in our case we have just three milestones in the process) and first of all if the correct process implementation is sufficient to guarantee the effectiveness of the verified safety functions.

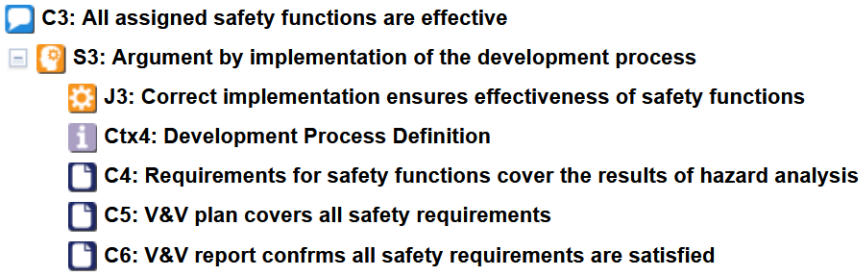


Figure 8. Reasoning step for claim C3 (NOR-STA notation)

The premises for the reasoning (claims C4, C5 and C6) refer to milestones of the development process. The claims say the milestones have been correctly implemented. The reasoning step is valid when the correct implementation of the process guarantees effectiveness of the implemented safety functions.

The line of reasoning of strategy S2 differs from the reasoning of the first strategy. We select a strategy adequate to the characteristics of the goal to be achieved. The strategies usually depend on the system lifecycle, regulations and the general engineering approach to the system development, deployment and operation. The most common types of argumentation strategies are:

- cause-effect inference rules where some objects or properties affect other objects or properties (an example is the strategy for the top claim: it says that effective safety functions affect the risk level of the system);
- decomposition by an object properties, architecture, by steps of a process or by other entities (an example is strategy S2 which is a decomposition based on the steps of the development process);
- calculations based on some measurement, statistics, estimation or analytical results (for example calculation of the value of a hazard probability and consequences);
- expert judgement based on past experience;
- rules based on regulations or required standards.

7 Evidential steps in the argument

The argument presented in Figure 7 / Figure 8 ends on claims C4, C5 and C6 which refer to the development process milestones. Depending on the process definition you may need to provide further argumentation but in our case the demonstration of the process work products would be sufficient. According to this we will develop evidential steps of the argument.

The structure of an evidential step is simple, it contains:

- base claims that are to be supported by evidence and which define acceptance criteria,
- the supporting evidence (in fact references to evidence items) and
- optionally context information when required.

For each of the claims we can provide evidence defined as work products in the development process.

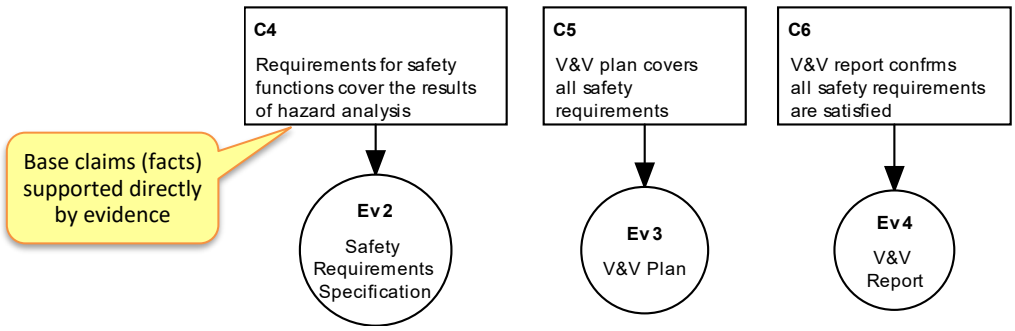


Figure 9. Evidential step of the argument for the development process implementation (GSN diagram)

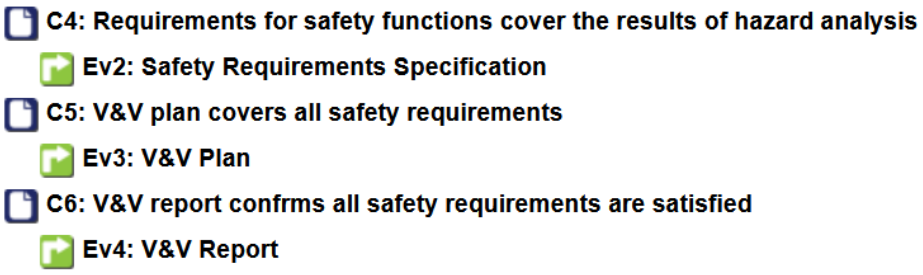


Figure 10. Evidential step of the argument for the development process implementation (NOR-STA notation)

Now the critical step is the definition of base claims. They should define acceptance criteria for the evidence. The names of base claims G4, G5 and G6 give the main acceptance criteria but the name is too short to give a complete list of requirements. Sometimes it is sufficient to refer to definitions, for example a milestone definition in the process description. But depending on the specifics of the supported reasoning it may be required to provide a detailed description of acceptance criteria for a base claim. We can consider two variants of a claim G5 definition.

Table 2. Variant 1 of claim C5 definition based on the development process definition

Label	C5
Name	V&V plan covers all safety requirements
Description	Verification and Validation (V&V) Plan satisfies the requirements of Development Process milestone “V&V Plan” and it follows the current recommendations of System Safety Department.

The first variant presented in Table 2 can be applied when the reasoning is based on some well documented processes which define precise criteria for work products. You may also refer to some checklists. When needed, you may add any requirements that are necessary to provide correct support in the argument. For example in variant 1 presented in Table 2 a requirement to follow some recommendations has been added.

It is important to define explicit requirements what evidence is required, not just name the title of evidence item and accept a document as it is.

When the list of requirements is long, it is usually an indicator that the assurance process is poorly defined and its documentation should be extended. In Table 3 you can find an example of a direct list of evidence acceptance criteria. In general you need a precise list of acceptance criteria, but such a list in the argument should be accepted only when the argumentation is for very specific and individual scenario. This may happen when the argument is developed for a new technology, novel engineering approaches or other areas not covered by defined system life cycle processes. However usually when you find a long list of acceptance criteria in an argument it is a sign that either the assurance process is not well defined or someone does not know it and have not referred to it. When you find a list of requirements line in Table 3 you should ask questions why this information is in the argument and not in system development process documentation.

Table 3. Variant 2 of claim C5 definition with a set of acceptance criteria

Label	C5
Name	V&V plan covers all safety requirements
Description	Verification and Validation (V&V) Plan satisfies the requirements: 1. it is based on the template defined in the Development Process, 2. it applies to the correct version of the device and other documents (requirements specification), 3. V&V Plan defines clear acceptance criteria for each test case, 4. V&V covers all safety requirements for all safety functions, 5. V&V covers all operation modes of the system, 6. V&V process is performed on all levels: unit, component, integration, system, 7. V&V process includes regression tests for all known bugs and failure modes, 8. V&V was applied to the assigned version of an item, no uncontrolled changes or patches were introduced, 9. All the results of V&V actions are documented, 10. bidirectional traceability are established between safety requirements and V&V test cases and the results, 11. V&V Plan specifies the tools, environment and test data used in the process,

8 Evidence

Assurance cases refer to evidence. Evidence item is an artifact containing information related to the object of the argument which we refer to it to reason about system properties. Evidence is an external element to the argument. Some evidence items may be documents under configuration management in the system life cycle.

An evidence item can be any artifact like a paper or electronic document, computer file or data, photo, video or any other information medium. Evidence can be created in the system development and operation process, can be a result of measurement, calculation, analysis, a formal proof or any other way.

Evidence is valid only when it describes some real properties related to the object of the argument in the past, present or estimations for the future. This is to be verified individually for each evidence item during the review of the supported claims.

Evidence items are elements external to the argument, they are not its components. The argument does not contain evidence but refers to them. Each evidence item should be accessible in a way defined in the argument. An electronic evidence URL address (Uniform Resource Locator) is often used as a reference. This allows to access the evidence when working with the argument.

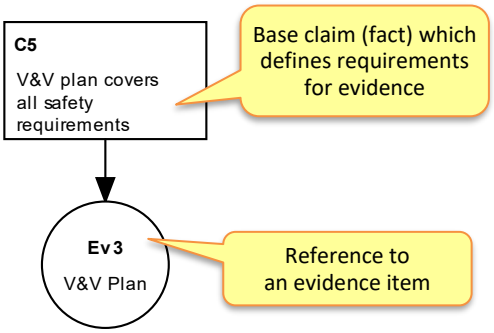


Figure 11. Evidence used to support a fact (base claim)

9 Assumptions

Sometimes in the argument we refer not only to available evidence but also to some assumptions we make without providing any evidence for them. For example you may notice that the maintenance of the device may have impact on the developed system and its safety, but maintenance activities are beyond control of device's manufacturer. To point it out we add an **assumption** element to the argument.

Any time you see a gap in the reasoning you may consider if there are any hidden assumption in the argument. When you identify such case you should specify it in an explicit way an as assumption. Any missing assumption makes the argument to be incomplete.

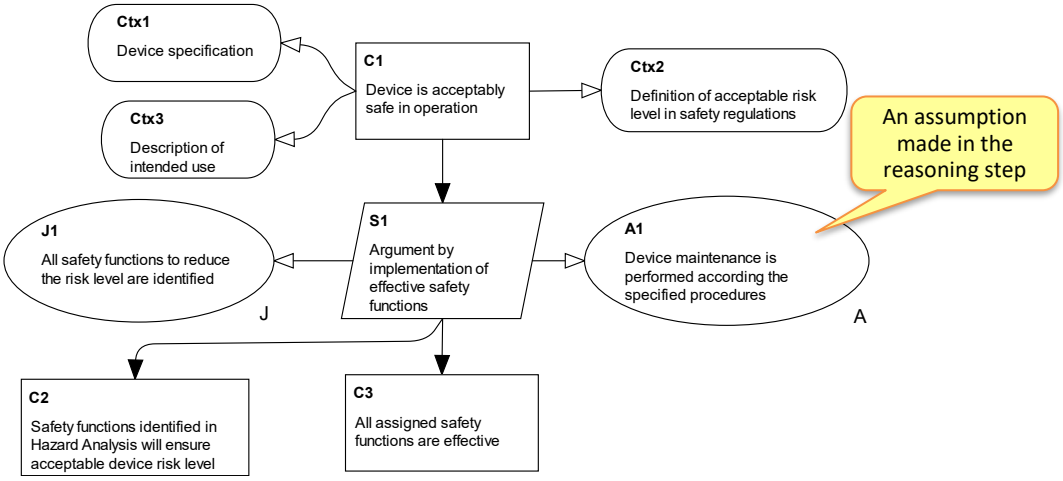


Figure 12. Assumption for a reasoning step (GSN diagram)

You may also use assumptions to represent some knowledge you have about the system and assume it to be true without requesting any evidence. An assumption may describe the context of the system, conditions of use or the environment. For example we may know that the system is intended to be used indoors and specify an assumption that the temperature during system operation will be above zero Celsius. When assumptions are used in this way they define limitations of the system.

All known assumptions should be specified in an explicit way

All assumptions should be verified during the argument review. The verification is specially important when you reuse parts of the argument as the context of the argument may change.

10 The complete argument

In this section we present the complete argument discussed in this Guide.

Please note that for a real system to produce a convincing argument you would probably define more argumentation steps (for example hazards are skipped in the presented argument) and an effective development process would require more steps and documents then referred in the argument.

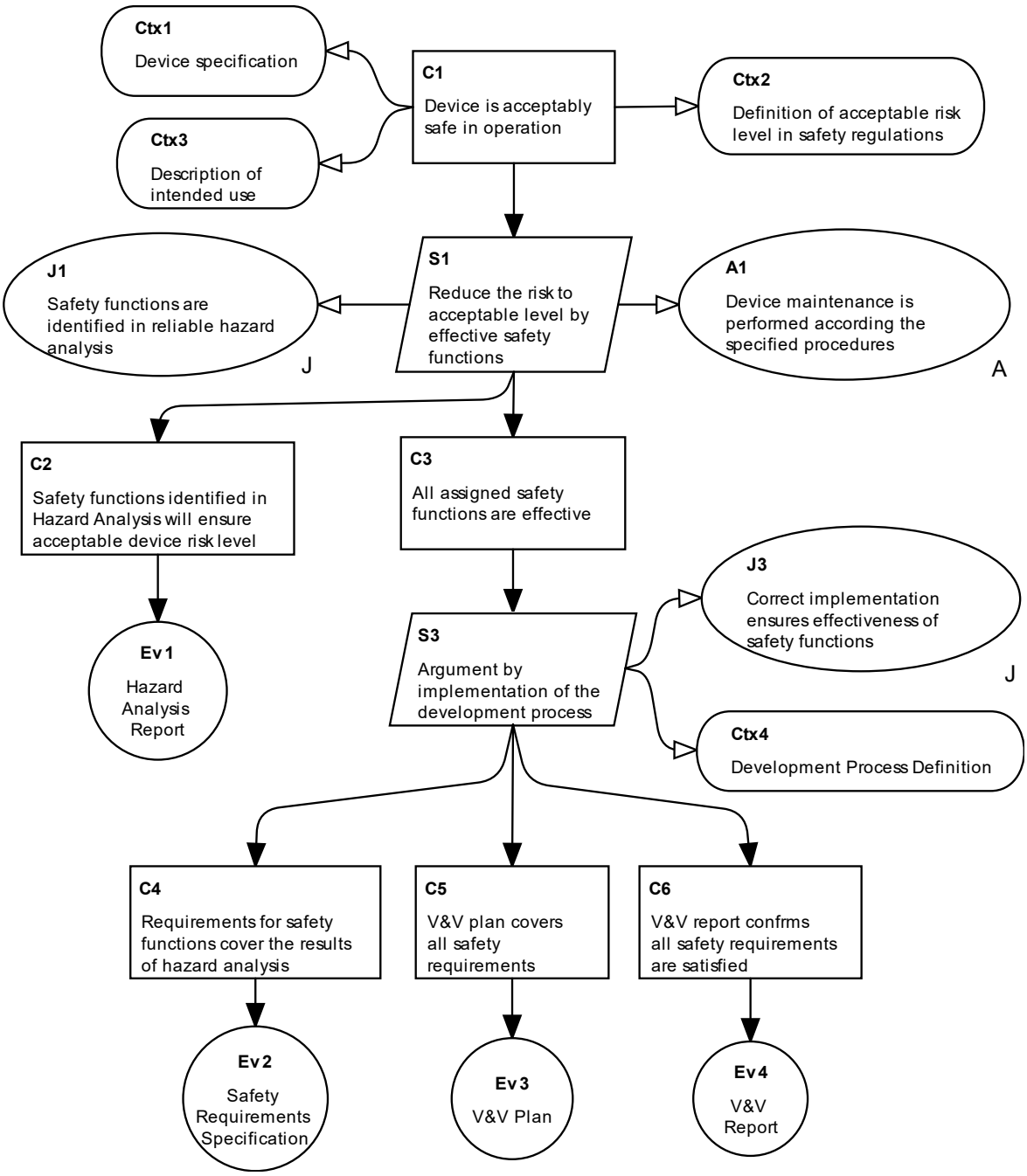


Figure 13. The complete argument (GSN diagram)

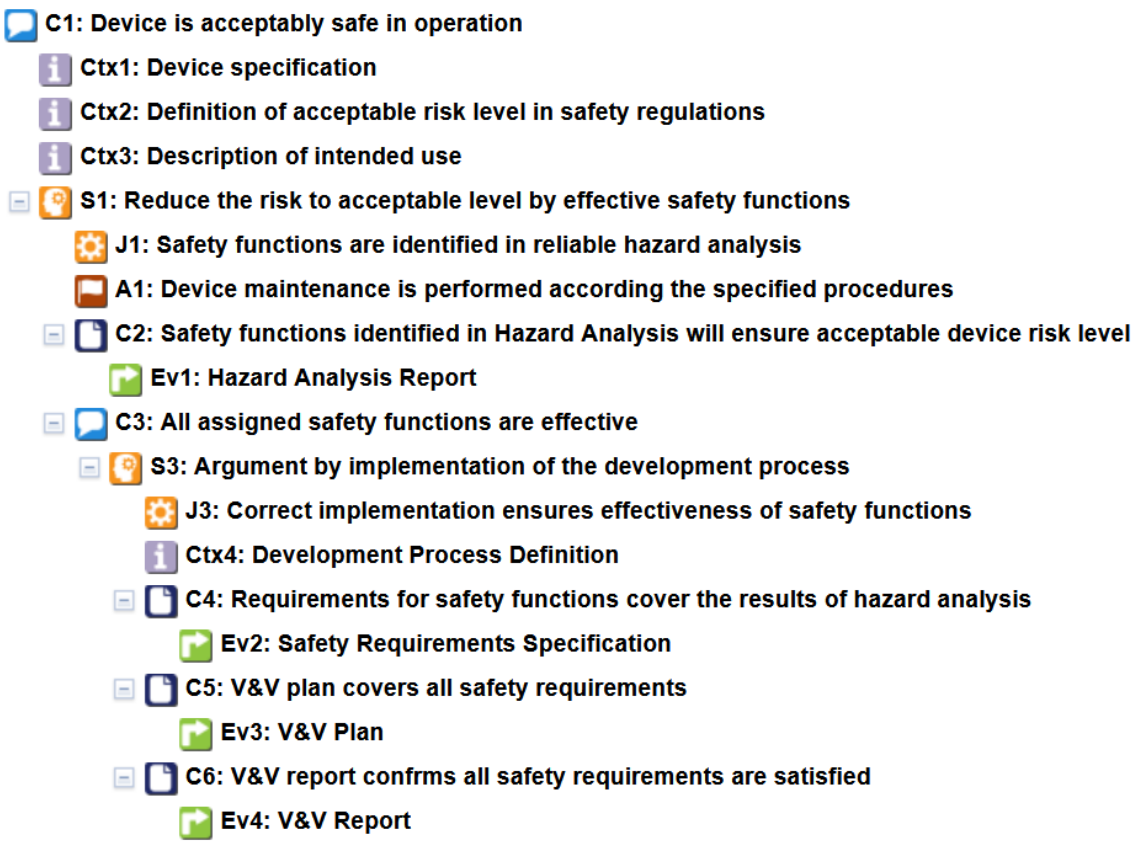


Figure 14. The complete argument (NOR-STA notation)

11 Argument structure definition

The general metamodel of NOR-STA argument structure is presented in Figure 15.

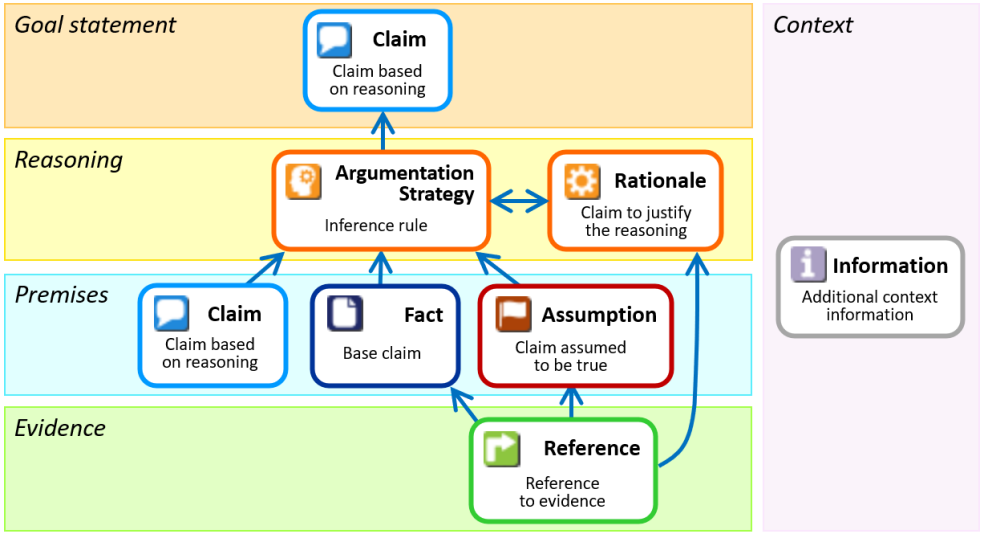


Figure 15. NOR-STA argument metamodel

Arrows on the diagram denote support. For example a claim is supported by a strategy. An element at the beginning of an arrow provides support for the element at its end.

1. The goal statement is a **claim**. It is a predicate (a true-false) statement which states the goal to be supported.
2. Reasoning is implemented by:
   a) an **argument strategy** which specifies the line of reasoning and
   b) **rationale** with justifies that the strategy is correct and applicable for a given claim and also that it is correctly applied.
3. A premise can be:
   a) a **claim** that is to be supported by another argumentation strategy and further premises or
   b) a **fact** (base claim) supported directly by evidence, or
   c) an **assumption** that we make in the argument.
4. Evidence which supports the premises:
   a) **references** to evidence artefact, usually documents but other media like photos or measurement data may also be used.
5. The context **information** may be provided to ensure precise interpretation of argument elements.

All the NOR-STA assurance case elements are listed in the table below:

Icon	Name	Definition
	Claim	A statement about some property that requires argumentation and evidence to demonstrate that the system satisfies it
	Argumentation Strategy	Strategy specifies the inference rule that uses the supporting premises to conclude that the claim is satisfied. Note: Strategy used to refute a claim (to conclude that it is not satisfied) is called a counter-argumentation strategy.
	Rationale	A statement that justifies validity of the reasoning set down for a given claim by the argumentation strategy

Icon	Name	Definition
	**Assumption**	A statement about some property, assumed to be true without any argument or evidence, usually assured by the environment, other systems or operators
	**Fact** (base claim)	A statement about some property supported directly by evidence Note: Facts are a type of claims that don't need any argumentation step and evidence is sufficient to demonstrate they are satisfied
	**Reference**	A reference to the evidence supporting the argument.
	**Information**	Additional description for the argument element that supplements its definition

NOR-STA argument structure should follow the rules described in the following subsections.

Each claim is supported by one or more strategies

NOR-STA notation requires each claim to be supported by at least one argumentation strategy. You cannot support a claim directly with other claims. Each reasoning step should be defined with the use of argumentation strategy and justified with a rationale.

More than one strategy for a claim can be defined to represent independent argumentations.

 Claim1: Software module is sufficiently reliable
 **Strategy1: Argue over module tests**
 **Strategy2: Argue over fixing all known bugs and regression tests**
 **Strategy3: Argue over formal proof**

Note: a base claim, that is, a claim supported directly by evidence is distinguished as a separate type “fact”.

A rationale is provided for each strategy

A rationale is to be specified for each strategy to justify that the strategy is correct and applicable for a given claim and also that it is was correctly applied.

 Claim1: Software module is sufficiently reliable
 Strategy1: Argue over module tests
 **Rationale1: Module testing process is reliable**

A rationale is a predicate (a true/false statement) like a claim. It can be supported by evidence or by arguments when necessary.

Argumentation strategy is supported by an arbitrary number of premises

A strategy should be supported by one or more premises. A premise can be a fact, an assumption or a claim.

 Claim1: Software module is sufficiently reliable
 Strategy1: Argue over module tests
 **Fact1: Tests reports show no errors**
 **Claim2: Tests cover all the scenarios described in the requirements**
 **Assumption1: Test team is competent**

Facts and assumptions are supported directly by references to the evidence

Facts and assumptions can be supported by evidence.

Fact1: Tests show no errors
Evidence3: Test report

Assumption1: Test team is competent
Evidence2: Test team members ISTQB certificates

Providing evidence for facts is mandatory in NOR-STA, while evidence for assumptions is optional.

Information element can be attached to any element

Additional information like a context data can be attached to any element using an information element. Such information element can be supported by references to documentation when needed.

Claim1: Software module is sufficiently reliable
Context1: Module design documentation
Reference1: Module requirements specification

Rationale can be supported by evidence or an argument

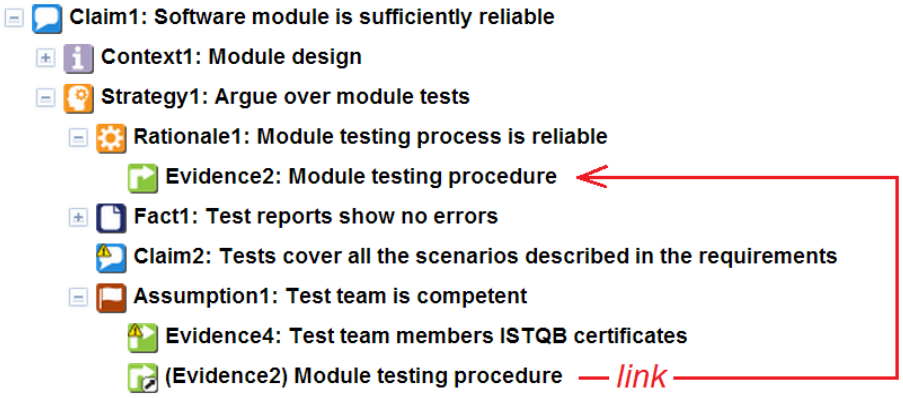
Rationale element can be supported either by an external evidence or by an explicit argument (a confidence argument) if a detailed argumentation is needed to build confidence in the rationale.

Rationale1: Module testing process is reliable
Evidence2: Module testing procedure

Linking argument elements

Some argument elements, for example references to evidence, can provide support for more than one argument element. Presenting assurance cases in a hierarchical way is a simplification and an argument is really a directed graph, not a hierarchical tree. We need a way to represent an element when it supports more than one element.

When one element is to be used more than once we can use links. Links are marked with a small black arrow in their icons:  . You can create links to any argument element except rationales.



12 How TRUST-IT relates to other assurance case approaches?

TRUST-IT method follows ISO 15026 standard for assurance cases and it is a specialized approach to deliver trustworthy arguments. It defines some additional restrictions and recommendations for the argument structure to achieve this goal. The main restrictions and rules specific to TRUST-IT are as follows:

1. In practice diagrams do not cover the full range of information of the argument. When you need more text to describe precisely an argument element you provide its description which is included in the assurance case report but not visible on the diagram. TRUST-IT method specifies some requirements about information that should be included in the element descriptions. This is in line with ISO 15026 standard which also sets requirements on the argument element definitions.
2. A strategy and rationale (justification) should be specified in an explicit way for each reasoning step. GSN allow claims to be supported directly by other claims. In some cases such direct support may be not clear for reviewers. TRUST-IT method requires that each reasoning step is precisely described with a strategy and rationale.
3. Rationale is a claim attached to a strategy that justifies it and can be extended with a separate argument when needed. This additional argument is called a confidence argument and its goal is to demonstrate that the way of reasoning is valid and it is implemented in the correct way. The same role plays "side claim" in the recent version of CAE notation (as published in Assurance 2.0).
4. TRUST-IT method distinguishes inferred and base claims. Inferred claims are supported by reasoning while the base claims are supported directly by evidence. The base claims are called "facts" in TRUST-IT method, because, when correct, they should be objective statements proved to be true based directly on evidence without any interpretation. TRUST-IT sets additional requirements on the way the facts (base claims) should be defined to avoid subjective interpretation of evidence.
5. TRUST-IT argument model is extended with the information about the assessment which includes value of the assessment, optionally confidence in the assessment and also comments which describe flaws in the argument and can provide guidelines how to improve it. The objective of the assessment extension is to enable a systematic process of reaching a strong and defensible argument. It also can be used as a documentation of this process. It facilitates the process of reaching consensus when two or more parties are involved in the assurance process or third-party assessment is conducted.

These are the main properties characteristic for the TRUST-IT model of the argument. As the method is focused on the process we should also note additional process-related requirements:

6. Change control and versioning are the required features of TRUST-IT method. The method assumes the final assurance case can be developed in a number of iterations which have to be managed and the argument developers may need to review past versions or recover them.
7. Collaboration in the assurance case process is possible when you can identify at least the users who make the assessment, give comments and introduce changes in the argument structure. The effective implementation of TRUST-IT approach requires management of authorship on the level of individual argument elements, their assessment and comments and also permission management system.

Finally TRUST-IT method is open for automation and the selected steps of the process can be automated.

The points listed above describe the main differences between TRUST-IT method and other approached to assurance cases.

This section does not apply to NOR-STA web application as the tool supports three argument models: TRUST-IT, GSN and ISO 15026.