A Categorization of GSN-based Safety Cases and Patterns

Yaping Luo, Zhuoao Li and Mark van den Brand
Mathematics and Computer Science, Eindhoven University of Technology,
P.O. Box 513, 5600 MB, Eindhoven, The Netherlands

Keywords: Goal Structural Notation, Safety Case, Safety Case Categorization, Safety Case Pattern, Safety Argument, Confidence Argument.

Abstract: Recently modeling techniques are introduced to support safety assessment. Goal Structural Notation is one of these modeling techniques, which can be used to facilitate the development of safety argumentation and create reusable safety argumentation models. Consequently, GSN-based safety cases are widely used to demonstrate the safety of systems in safety-critical domains. Due to the amount of manual work, constructing a safety case is usually time-consuming. Moreover, the re-usability of GSN-based safety cases is limited. To address this, safety case patterns are introduced to support safety case reuse. As more and more GSN-based safety cases and patterns are designed with different goals in different contexts, it becomes hard to identify a reusable safety case or pattern. In this paper, we carried out a study on the categorization of existing GSN-based safety cases and patterns. As a result, a number of high cited publications are selected and studied. Finally a categorization of GSN-based safety cases is proposed. A clear categorization of GSN-based safety cases can be used to identify similar safety cases or patterns and facilitate safety case reuse.

1 INTRODUCTION

A safety case is a well-structured argument for justifying that a system is safe. In (Bishop and Bloomfield, 1998), a safety case is defined as: "A documented body of evidence that provides a convincing and valid argument that a system is adequately safe for a given application in a given environment". In some international safety standards, explicit safety cases are required for safety-critical systems. For example, ISO 26262 (ISO26262, 2011), for the automotive domain, stimulates the use of safety cases to demonstrate the product safety (Safety Case Repository, 2013). Besides, MOD Def Stan 00-55 (MOD, 1997) for safety-critical software in defense equipment requires producing safety cases with explicit safety requirements.

Typically, safety cases are represented in free text, but in this way, the structure of the safety cases might be unclear, which allows for inconsistencies and confusion (Luo et al., 2015a) (Luo et al., 2015b). To address this, modeling techniques are introduced to facilitate safety case construction and to increase the understandability and confidence in the claimed safety assurance (Safety Case Repository, 2013). For example, techniques originally from model-driven development are used for modeling safety standards, and representing concepts in safety cases, such as ontologies, and SBVR models. Goal Structural Notation (GSN) is introduced as a graphical modeling approach for safety case construction (Kelly and Weaver, 2004). The details of GSN are described in Section 2. With the increase of safety-critical software or systems, such as cars, more and more GSN-based safety cases are developed. The re-usability of GSN-based safety cases becomes another challenge. People want to reuse safety case whenever it is possible. Informal reuse of safety case elements occurs, like 'Copy and Paste' of the textual safety case documents between projects. A number of problems with informal reuse are listed in (Kelly and McDermid, 1998). For example, it may cause inappropriate reuse, lack of traceability, or lack of consistency. To prevent these problems, safety case patterns are introduced as an approach to reuse of common structures of safety cases.

As a number of GSN-based safety cases and patterns has been developed, when constructing a safety case, engineers can reuse or build their safety case upon existing ones. In order to accomplish this, a large number of safety cases or patterns from different sources need to be collected, then the similar ones for reuse have to be identified. This process can be very time-consuming. Therefore, categorizing those safety cases or patterns into smaller groups is promising.

In this paper we propose a categorization of GSN-

Luo, Y., Li, Z. and Brand, M.
A Categorization of GSN-based Safety Cases and Patterns.
DOI: 10.5220/0005734305090516
In *Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2016)*, pages 509-516
ISBN: 978-989-758-168-7

based safety cases and patterns, for this we selected and studied a number of high cited publications [1]. The publications are divided into two groups: one group for categorizing, and one group for the validation of the categorization. As safety case patterns can be used as templates for creating safety cases, patterns represent general characteristics of safety cases. Thus, we collected existing safety case patterns from a group of papers for validation.

The aim of our research is to form a categorization based on existing studies, and validate the categorization by applying it to the widely-used safety case patterns. If existing safety cases or patterns are categorized into groups according to their original goals, then engineers facing a particular safety requirement to fulfill, they will be able to look into a small specific collection of safety cases or patterns for the reusable ones. In this way, the efficiency of the reuse process can be improved. The structure of this study is as follows. Section 2 introduces the background information about GSN (Goal Structuring Notation) and safety case pattern. Section 3 provides an overview of a categorization of safety cases and definitions of each type argument. Section 4 discusses the validation results by categorizing existing safety case patterns. Finally, a conclusion of this research and the expected future work are mentioned.

2 BACKGROUND

2.1 GSN and Extension

Goal Structuring Notation (GSN) is a graphical notation which is widely recommended for modeling safety cases (Kelly and Weaver, 2004). It provides a clear and well-structured argument in terms of basic graphical elements, such as goals, solutions and strategies. The primary elements of the standard GSN are introduced in Figure 1.

When a GSN-based safety case is successful, people tend to define a template of the safety case for reusing, e.g. safety case patterns (Kelly and McDermid, 1997). As safety case pattern requires more flexibility and complexity on the structure, the standard GSN has been extended with several elements and entities (Figure 2). The detailed information of the elements can be found in the GSN Community Standards Version 1 (GSN Community Standard, 2011). Only the sixth element (Assurance Claim Point, ACP) is not described in the GSN Community Standards. This arrow is introduced in (Hawkins et al., 2011), which means a confidence argument should be indicated for this connection.

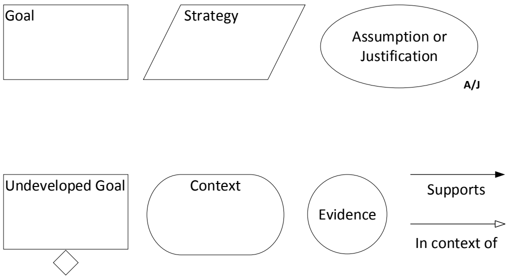

Figure 1: Primary elements of the Goal Structuring Notation.

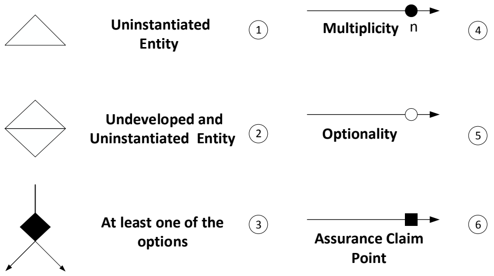

Figure 2: Extension of GSN to support safety case pattern.

2.2 Safety Case Pattern

As mentioned before, the safety case pattern approach supports the reuse of safety cases by identifying and recording reusable arguments. The format of a safety case pattern includes two parts: description of the design and the structure represented in GSN. The description of a safety case pattern is often written in the form of a table described in (Kelly and McDermid, 1997). In this table, a pattern developer provides all related information about the pattern, such as the intent, motivation, applicability and implementation. This information helps the reviewers to understand and reuse the pattern.

Safety case patterns facilitate modeling of safety cases. The developer only needs to consider the variables in each element of the pattern and concentrate on collecting corresponding evidences and contexts. Using safety case patterns to identify and record the reusable arguments will improve the re-usability of safety cases and save a lot of time and resources.

3 SAFETY CASE CATEGORIZATION

Research and development on safety cases has taken

[1] In the bibliography the number of citations per paper based on Google Scholar is explicitly given.

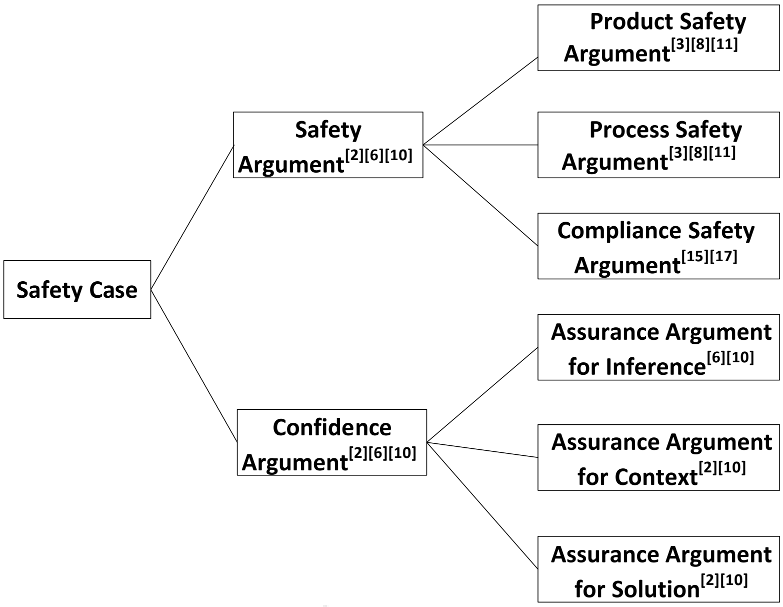
Safety Case
Safety Argument[2][6][10]
Confidence Argument[2][6][10]
Product Safety Argument[3][8][11]
Process Safety Argument[3][8][11]
Compliance Safety Argument[15][17]
Assurance Argument for Inference[6][10]
Assurance Argument for Context[2][10]
Assurance Argument for Solution[2][10]

Figure 3: A categorization of safety cases.

more than two decades, but there is still no specific categorization of safety cases. A number of publications (Hawkins et al., 2011) (Birch et al., 2013) (OPENCOSS, 2013) (Habli and Kelly, 2006) indicate a simple classification of safety cases. Therefore, we carry out a study based on those publications to obtain an explicit categorization of safety cases. In the following section, a GSN-based safety case categorization are discussed.

3.1 Safety Case Categorization Overview

An overview of our proposed safety case categorization is shown in Figure 3. At the first level, safety cases are categorized in two types: Safety Argument and Confidence Argument. This separation is derived from (Hawkins et al., 2011). Safety argument is the argument that purely explains how the evidence supports the claims of the acceptable safety of a system. Confidence argument is separated from safety argument, because it serves to explain the confidence of a safety argument. (Ayoub et al., 2012) and (Denney et al., 2011) propose approaches for systematically constructing confidence arguments and evaluating the uncertainties of safety arguments. For most of the existing safety cases, the confidence argument is always included in the safety argument. When the distinction between safety and confidence arguments is made, a safety case developer will have a clear direction on the safety case construction steps. For safety case reviewers or assessors, the distinction will help them to understand a safety case better and identify these weakly supported aspects (Hawkins et al., 2011).

3.2 Safety Argument

In a safety argument, the goal is to argue that system itself or the development of the system is acceptably safe. For example, a risk-based safety argument may contain the identification and mitigation of hazards associated with the system. Every assertion and evidence in this safety argument will have a direct role to the claims about the hazards. This kind of argument is usually deterministic. Depending on the content of claims, the safety argument can be further categorized in three types: product safety argument (Birch et al., 2013) (Habli and Kelly, 2006) (ISO26262, 2011), process safety argument (Birch et al., 2013) (Habli and Kelly, 2006) (ISO26262, 2011), and compliance safety argument (Kelly, 1999) (OPENCOSS, 2013).

3.2.1 Product Safety Argument

If the top-claim in a safety argument focuses on safety characteristics of a specific product, this argument is classified as a product safety argument (or product-based safety argument). When a claim in this argument is not trivial to be asserted, it will be split into sub-claims. Finally, product-based evidence will be provided for the bottom-level safety claims. An example of this type of safety argument can be an argument constructed to demonstrate the satisfaction of

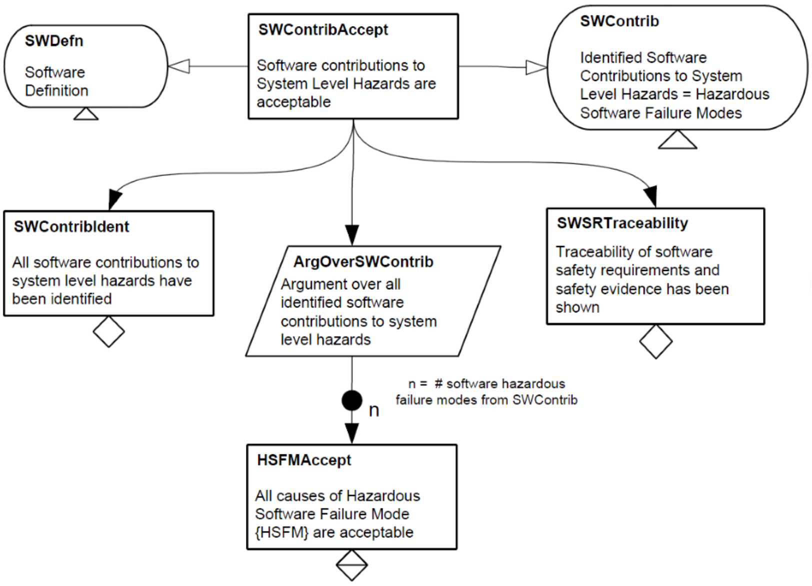
SWDefn
Software Definition
SWContribAccept
Software contributions to System Level Hazards are acceptable
SWContrib
Identified Software Contributions to System Level Hazards = Hazardous Software Failure Modes
SWContribIdent
All software contributions to system level hazards have been identified
ArgOverSWContrib
Argument over all identified software contributions to system level hazards
SWSRTraceability
Traceability of software safety requirements and safety evidence has been shown
n = # software hazardous failure modes from SWContrib
n
HSFMAccept
All causes of Hazardous Software Failure Mode {HSFM} are acceptable

Figure 4: An example of product safety argument pattern (Weaver, 2004, Page 208).

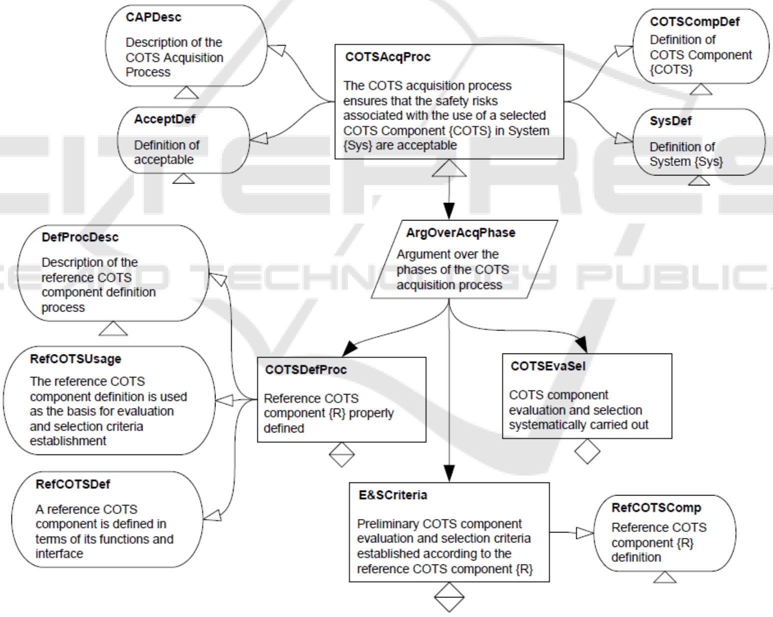
CAPDesc
Description of the COTS Acquisition Process
AcceptDef
Definition of acceptable
COTSAcqProc
The COTS acquisition process ensures that the safety risks associated with the use of a selected COTS Component {COTS} in System {Sys} are acceptable
COTSCompDef
Definition of COTS Component {COTS}
SysDef
Definition of System {Sys}
ArgOverAcqPhase
Argument over the phases of the COTS acquisition process
DefProcDesc
Description of the reference COTS component definition process
RefCOTSUsage
The reference COTS component definition is used as the basis for evaluation and selection criteria establishment
RefCOTSDef
A reference COTS component is defined in terms of its functions and interface
COTSDefProc
Reference COTS component {R} properly defined
COTSEvaSel
COTS component evaluation and selection systematically carried out
E&SCriteria
Preliminary COTS component evaluation and selection criteria established according to the reference COTS component {R}
RefCOTSComp
Reference COTS component {R} definition

Figure 5: An example of process safety argument pattern (Ye, 2005, Figure 40).

the safety requirements in the hazard analysis phase. This argument will mainly rely on the generated work products, which can be used as supporting evidence. Therefore, this argument is a typical product safety argument. Normally, the product safety argument is the crucial part of a completed safety case. Other types of safety arguments are built around it.

A pattern of this type of safety argument can be found in (Weaver, 2004) and is shown in Figure 4. This pattern can be used to argue that the software contributions to system level hazards are acceptable. The instantiation of this pattern will be a safety argument on a specific product, for example a safety argument on Hazardous Software Failure Modes.

3.2.2 Process Safety Argument

When the top-claim in a safety argument focuses on the quality of the development process, then this argument can be classified as a process safety argument. For example, if an argument states that tools or methods used in the development process satisfy the corresponding

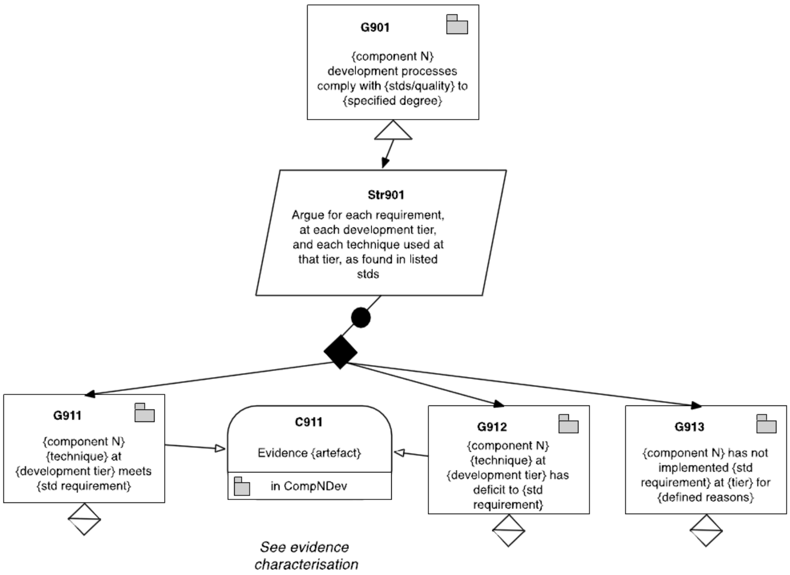
G901
{component N} development processes comply with {stds/quality} to {specified degree}
Str901
Argue for each requirement, at each development tier, and each technique used at that tier, as found in listed stds
G911
{component N} {technique} at {development tier} meets {std requirement}
C911
Evidence {artefact}
in CompNDev
G912
{component N} {technique} at {development tier} has deficit to {std requirement}
G913
{component N} has not implemented {std requirement} at {tier} for {defined reasons}
See evidence characterisation

Figure 6: An example of compliance safety argument pattern (OPENCOSS, 2013, Figure 32).

safety requirements, it will be referred to as a process safety argument. A process safety argument focuses on the implicit development process assumptions underlying the product-based argument. Thus, this type of argument is strongly connected with a product safety argument.

An example of process safety argument pattern is shown in Figure 5 (Ye, 2005). The pattern is used to argue the safety of a Commercial-Off-The-Shelf (COTS) component use from a risk management perspective. It shows how the safety risks associated with the use of a COTS component are identified and addressed throughout the COTS acquisition process. The strategy of this argument is to decompose the acceptability of the COTS acquisition process to three essential phases: defining a reference COTS component, establishing the evaluation and selection criteria, and conducting the COTS component evaluation and selection.

3.2.3 Compliance Safety Argument

When the top-claim in a safety argument points explicitly to a specific safety standard or regulation, this argument can be classified as a compliance safety argument. The goal of this type argument is to demonstrate that the development process of a system or the system itself is in compliance with the target standards or regulations. For example, for a top-claim "the hazard analysis process of system X complies with ISO 26262", a safety argument will need to be provided to argue the compliance with ISO 26262 standard. Then necessary evidence will be required to support such argument. This type of argument contains the compliance statements underlying the product-based and process-based argument.

A pattern for constructing a compliance safety argument is shown in Figure 6. This pattern is designed for arguing how well the development process of one component complies with a given standard.

3.3 Confidence Argument

A confidence argument is used to explicitly address uncertainties in inference rules and help explain why there is sufficient confidence in a safety argument. In other words, a confidence argument documents the justification for confidence in a safety argument. In the past, confidence argument is implicitly included in the safety argument. The distinction between confidence and safety argument is proposed in (Hawkins et al., 2011). The goal of that research is to manage the confidence part explicitly. In this way, the clarity of a safety case can be improved, which facilitates safety case construction and review process. When building a safety argument, a number of assertions will be made. These assertions represent the principles of the safety argument. For different types of assertions, a confidence argument can be further categorized in three types: assurance argument for inference, context and solution. Some examples of different types of confidence argument can be found in (Hawkins et al., 2011).

3.3.1 Assurance Argument for Inference

In a safety argument, if a top-level claim is too extensive to be argued, the claim will be split into several sub-claims through inference. The reason or method

Table 1: Categorization of the safety case patterns in each paper.

Publications	Product	Process	Compli-ance	Asserted Inference	Asserted Context	Asserted Solution	Mixed
(Kelly, 1999)	4	1	3	-	-	-	3
(Weaver, 2004)	12	-	-	-	-	-	-
(Ye, 2005)	9	11	-	-	-	1	-
(Alexander et al., 2007)	9	-	-	-	-	-	-
(Hawkins and Kelly, 2008)	1	4	-	-	-	-	-
(Robert and Ibrahim, 2010)	4	1	-	-	-	-	-
(Conmy and Bate, 2014)	-	-	-	-	-	1	-
(OPENCOSS, 2013)	3	4	2	-	-	-	-

for this inference is documented in a strategy element. Then to gain assurance in the strategy, a confidence argument is needed to demonstrate why the asserted inference should be trusted. We call this type of confidence argument "Assurance Argument for Inference". An assurance argument for inference is placed between the parent claim and its strategy or sub-claims.

3.3.2 Assurance Argument for Context

Usually, contextual information contains the information that is referred to in the claim or strategy. For example, when a strategy states “argument over all system hazards”, then a context pointing to the corresponding hazard list will be added. When a context is introduced into a safety argument, the appropriateness and trustworthiness of that context should be taken into account. Thus a confidence argument for the context is necessary. We call this type of confidence argument “Assurance Argument for Context”. An assurance argument for context can be given between the context and its corresponding strategy or claim to represent the reliability of its source.

3.3.3 Assurance Argument for Solution

Assurance Argument for Solution is a kind of confidence argument for showing the confidence on evidence. When evidence is provided as a solution to an argument, it need to be assured that the evidence is sufficient to support the claim. Therefore, an assurance argument for solution can be added for this purpose. The assurance argument is given between solution and its corresponding claim.

4 VALIDATION

Since the use of safety case patterns as a method of documenting and reusing safety case structures was pioneered by Kelly in 1999, the development on safety case patterns has made great progress. Up to now, a large number of safety case patterns have been developed for different domains, for instance software (Weaver, 2004), automotive (Robert and Ibrahim, 2010) and COTS Components (Ye, 2005). Safety cases can be built based on these safety case patterns. Therefore, by categorizing safety case patterns, the safety cases derived from these patterns can also be categorized. As safety case patterns have the common characteristics of safety cases, we choose to apply our categorization on a number of safety case patterns instead of a random set of safety cases. We selected 8 existing paper which includes specific safety case patterns (Kelly, 1999) (Weaver, 2004) (Ye, 2005) (Alexander et al., 2007) (Hawkins and Kelly, 2008) (Robert and Ibrahim, 2010) (Hawkins et al., 2011) (OPENCOSS, 2013). In total, 73 safety case patterns have been collected from those papers. Those safety case patterns have been categorized according to their safety goals. Table 1 shows the results of this categorization. Note that, some safety case patterns have more than one type of safety argument, then they are labeled as ‘Mixed’.

In (Kelly, 1999), eleven patterns are described for both domain specific and domain independent goals. These patterns are categorized into four product safety arguments, one process safety argument, three compliance safety arguments, and three mixed types. The designed patterns in (Weaver, 2004) and (Alexander et al., 2007) are domain specific and product-based, they are classified into patterns for product safety arguments. Besides, The patterns in (Ye, 2005) are classified into nine for product safety arguments, one for process safety arguments, and one for assurance arguments for solution. Those patterns are specifically developed to support the safety of a system which includes COTS component. The patterns in (Hawkins and Kelly, 2008) and (Robert and Ibrahim, 2010) are introduced as a part of a safety case pattern catalog. In (Hawkins and Kelly, 2008), a software safety case pattern catalog is described for constructing arguments on software safety. In (Robert and Ibrahim, 2010), an automotive safety case pattern catalog is designed according to

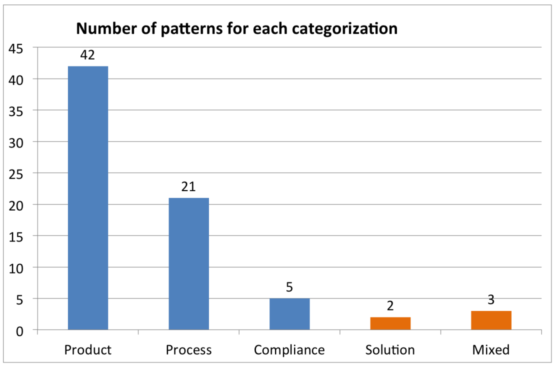

Figure 7: Statistical results of Categorization: for asserted inference and asserted context, there is no confidence argument patterns found in those papers.

the ISO 26262 standard. These catalogs are built upon existing work or good practices in the safety domain. In total, five patterns for product safety arguments and five patterns for process safety arguments are found in these two catalogs. A confidence argument pattern for describing evidence, and its provenance and quality is provided in (Conmy and Bate, 2014), therefore it is categorized as an assurance argument for solution. Furthermore, strategies for modular arguments has been discussed in (OPENCOSS, 2013). The safety argument is modularized to separate concerns for different purposes. The goal of this is to support safety case reuse between different safety domains. For each module, template arguments (argument patterns) has been provided. From those template arguments, three patterns for product safety arguments, four patterns for process safety arguments, and two patterns for compliance safety arguments are found.

Figure 7 shows how the collected safety case patterns distributed over all the types. We can see that product and process safety arguments form the majority. There are several reasons for this result. Firstly, safety standards themselves are product-oriented or process-oriented, therefore, a part of compliance argument has already been covered by product or process safety argument. Secondly, confidence safety argument is a new topic proposed in recent years. The development of confidence safety argument is not mature as safety argument. Only two confidence arguments on asserted solution are found in those papers. The reason for this could be: for the collected patterns, the motivations for inference and context have already been documented in the relevant GSN elements, In other words, the confidence arguments on asserted inference and context are implicitly covered by safety argument. Moreover, the appearance of these two types of confidence arguments is low, therefore, the safety patterns for them are seldom created. Finally, as most of safety arguments are classified as product or process safety arguments, there is a possibility that more classifications can be introduced for these two types safety argument. Then the depth of the categorization can be increased, and more specific types of safety argument can be identified.

5 CONCLUSIONS

In this paper, we presented a safety case categorization according to several high cited publications. For each of the classification, we discussed its definition and common characteristics that should be considered by safety case writers and readers. Then we collected a number of safety case patterns from another group of papers to validate our categorization.

The results (Figure 7) show that most existing safety cases focus on safety argument, especially product and process argument. It is possible to classify those two classifications further for specific domains. As confidence safety argument is a new research topic, there are still a lot of room for development.

Threats to Validity. There are some threats to validity related to this study. Firstly, the number of selected publications is restricted. Thus we chose papers according to the number of citations. Secondly, most of selected high cited papers are from University of York. They have a lot experience in this domain and they published a large number of papers with high effect on the GSN-based safety case research and practical community. Finally, low cited papers without concrete GSN-based safety cases or patterns are excluded from our study. Because they do not provide new insight.

Future Work. As future work, we plan to improve the current categorization by increasing its depth and accuracy. Besides, we would like to use categorization in some industrial case studies to facilitate modeling safety cases, and support safety case modularity and reuse.

REFERENCES

Alexander, R., Kelly, T., Kurd, Z., and McDermid, J. (2007). Safety Cases for Advanced Control Software: Safety Case Patterns. Technical report, DTIC Document. Cited by 16.

Ayoub, A., Kim, B., Lee, I., and Sokolsky, O. (2012). A Systematic Approach to Justifying Sufficient Confidence in Software Safety Arguments. In *Computer Safety, Reliability, and Security*, pages 305–316. Springer. Cited by 7.

Birch, J., Rivett, R., Habli, I., Bradshaw, B., Botham, J., Higham, D., Jesty, P., Monkhouse, H., and Palin, R. (2013). Safety Cases and Their Role in ISO 26262 Functional Safety Assessment. *Computer Safety, Reliability, and Security Lecture Notes in Computer Science Volume 8153, pp 154-165*. Cited by 11.

Bishop, P. and Bloomfield, R. (1998). A Methdology for Safety Case Development. *Industrial Perspectives of Safety-critical Systems, P194-203*. Cited by 201.

Conmy, P. and Bate, I. (2014). Assuring Safety for Component Based Software Engineering. In *2014 IEEE 15th International Symposium on High-Assurance Systems Engineering (HASE)*, pages 121–128. Cited by 3.

Denney, E., Pai, G., and Habli, I. (2011). Towards Measurement of Confidence in Safety Cases. In *2011 International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 380–383. IEEE. Cited by 26.

GSN Community Standard (2011). GSN Community Standard: Version 1; November 2011, (c) 2011 Origin Consulting (York) Limited. http://www.goalstructuringnotation.info/documents/GSN_Standard.pdf/.

Habli, I. and Kelly, T. (2006). Process and Product Certification Arguments: Getting the Balance Right. *ACM SIGBED Review*, 3(4):1–8. Cited by 32.

Hawkins, R. and Kelly, T. (2008). A Software Safety Argument Pattern Catalogue. *Department of Computer Science, The University of York*. Cited by 13.

Hawkins, R., Kelly, T., Knight, J., and Graydon, P. (2011). A New Approach to Creating Clear Safety Arguments. *Advances in Systems Safety, pp 3-23*. Cited by 63.

ISO26262 (2011). ISO: ISO 26262 Road Vehicles – Functional Safety. ISO Standard.

Kelly, T. (1999). *Arguing Safety: A Systematic Approach to Managing Safety Cases*. PhD thesis, University of York. Cited by 295.

Kelly, T. and McDermid, J. (1997). Safety Case Construction and Reuse using Patterns. In *SafeComp 97*, pages 55–69. Springer. Cited by 107.

Kelly, T. and McDermid, J. (1998). Safety Case Patterns - Reusing Successful Arguments. In *IEEE Colloquium on Understanding Patterns and Their Application to Systems Engineering (Digest No. 1998/308)*, pages 3/1–3/9. cited by 41.

Kelly, T. and Weaver, R. (2004). The Goal Structuring Notation - A Safety Argument Notation. *Proc. of Dependable Systems and Networks 2004 Workshop on Assurance Cases*. Cited by 257.

Luo, Y., van den Brand, M. G. J., Engelen, L., and Klabbers, M. (2015a). A Modeling Approach to Support Safety Assurance in the Automotive Domain. In *Progress in Systems Engineering*, volume 1089, pages 339–345. Springer International Publishing.

Luo, Y., van den Brand, M. G. J., and Kiburse, A. (2015b). Safety Case Development with SBVR-based Controlled Language. In *Proceedings of Third International Conference on Model-Driven Engineering and Software Development*.

MOD (1997). Defence Standard 00-55 Part 1. http://www.software-supportability.org/Docs/00-55_Part_1.pdf.

OPENCOSS (2013). OPENCOSS: Deliverable D5.3 - Compositional certification conceptual framework (report). http://www.opencoss-project.eu/node/7.

Robert, P. and Ibrahim, H. (2010). *Assurance of Automotive Safety–A Safety Case Approach*. Springer. Cited by 16.

Safety Case Repository (2013). Safety Case Repository. http://dependability.cs.virginia.edu/info/Safety_Cases:Repository.

Weaver, R. (2004). *The Safety of Software - Constructing and Assuring Arguments*. PhD thesis, Department of Computer Science, University of York. Cited by 102.

Ye, F. (2005). *Justifying the Use of COTS Components within Safety Critical Applications*. PhD thesis, Department of Computer Science, University of York. Cited by 21.