A Pattern to Argue the Compliance of System Safety Requirements Decomposition

A. DE OLIVEIRA, University of São Paulo
R. T. V. BRAGA, University of São Paulo
P. C. MASIERO, University of São Paulo
I. HABLI, The University of York
T. KELLY, The University of York

Safety case is a structured argument aimed to argue the system is acceptably safe to operate in a specific environment. Safety cases have been successfully used as documentation to support the certification process of safety-critical systems. Safety case patterns have been used to document recurrent safety argumentation solutions that have evolved over time by providing a simple and efficient solution to a particular problem. There are in the literature safety case patterns providing recurrent solutions for arguing the safety of product and process, and the compliance of development processes with safety standards. These patterns may require information provided by safety analysis to be instantiated. The decomposition of system safety requirements (SSRs) throughout component-level failures is one of the activities of the system safety analysis. Existing model-based safety analysis tools provide the capability of automatically allocate SSRs to component failures. In this context, there is a need for an structured safety argument arguing the conformance of the allocation of SSRs provided by these tools with requirements allocation rules prescribed by safety standards as ISO 26262 and ARP 4754A. In this paper, we propose safety case pattern describing a reusable argument structure to argue the compliance of the requirements allocation with the assignment rules prescribed by safety standards. The pattern addresses to support safety analysts in developing safety arguments to justify sufficiency and compliance of the requirements decomposition with the target safety standard. This pattern was applied in the design of arguments of two systems from automotive and avionics domains.

Categories and Subject Descriptors: **H.1.2 [Models and Principles]**: User/Machine Systems—*Human Information Processing*; General Terms: Software Engineering

Additional Key Words and Phrases: Safety requirements, safety standard, system hazard, component failure mode.

ACM Reference Format:

Oliveira, A. L., Braga, R. T. V., Masiero, Habli, I., Kelly, T., C. 2014. A Pattern to Argue the Compliance of System Safety Requirements Decomposition. 10th Latin American Conference on Pattern Languages of Programs.

1. INTRODUCTION

The aim of a safety case is to argue that a system is acceptably safe to operate in a specified context. Such argument demonstrates how the available evidence can be interpreted as compliance with the applicable safety objectives. Safety Cases are expressed by means of Goal Structuring Notation (GSN) (Kelly, 2003). GSN provides the following notation elements: Goal, Solution, Strategy, Assumption/Justification, and Context. These elements are placed together to form a goal structure.

The purpose of a goal structure is to show how a **goal** is broken into **sub-goals**, and supported by evidence (**solutions**) whilst making clear the **strategies** adopted, the rationale by means of **assumptions** and **justifications**, and the **context** in which the goals are stated (Kelly, 2003). Safety Case patterns are an attempt to capture recurrent safety argumentation solutions that have evolved over time by providing a simple and efficient solution to a particular problem, whether in the execution of a safety process or the construction of a particular safety argument. Safety Case patterns are described using the GSN patterns extension that supports structural and entity abstractions (Kelly and McDermid, 1997).

The development of safety-critical systems like automotive and avionics require safety analysis activities to identify the system hazards, their causes, and to allocate safety requirements to eliminate or

This work is supported by the CNPq, process number 152693/2011-4.
Author's address: A. L. De Oliveira, University of São Paulo, São Carlos, Brazil; email: andre_luiz@icmc.usp.br; R. T. V. Braga, University of São Paulo, São Carlos, Brazil; e-mail: rtvb@icmc.usp.br; P. C. Masiero, University of São Paulo, São Carlos, Brazil; email: masiero@icmc.usp.br; I. Habli, The University of York, York, United Kingdom; e-mail: ibrahim.habli@york.ac.uk; T. Kelly, The University of York, York, United Kingdom; e-mail: tim.kelly@york.ac.uk

minimize the hazard effects. Safety standards like ISO 26262 (ISO, 2011) for automotive, and ARP 4754A/ED-79A (EUROCAE, 2010) for avionics provide a set of assignment rules to decompose the safety requirements allocated to system hazards throughout the failures in components of the system architecture. Arguing how the safety requirements allocated to component failures fulfill the requirements allocated to system-level hazards in compliance with a target safety standard can be used as evidence in a certification process.

In this document, we propose an argument pattern to argue the compliance of requirements allocated to system-level hazards and their contributing components failures with the requirements assignment rules prescribed by safety standards. This pattern aims to support safety analysts in generating safety arguments arguing the safety of critical systems from the safety analysis information provided by model-based safety analysis tools like HiP-HOPS (Azevedo et al., 2014). Next section presents the pattern to argue the Compliance of System Safety Requirements Decomposition. The description of the GSN notation elements used to describe the pattern can be found in the appendix.

2. SYSTEM SAFETY REQUIREMENTS DECOMPOSITION COMPLIANCE ARGUMENT PATTERN

2.1 Intent

To provide a framework for justification of sufficiency of the safety requirements allocated to failures in components of the system architecture so that they fulfill the requirements allocated to system-level hazards. This pattern aims to provide an structure to argue that such requirements allocation complies with requirements assignment rules prescribed by safety standards like ISO 26262, and ARP 4754A.

The pattern aims to provide elements to verify whether the requirements allocated to component failures and hazards comply with the safety standard assignment rules. This pattern addresses the provision of a compliance argument structure which can be used as certification evidence for a safety-critical system.

2.2 Also Known As

System Safety Requirements Decomposition Compliance, Requirements Compliance, and Requirements Correctness.

2.3 Problem

How can we argue that the requirements allocated to system hazards and their contributing component failures comply with the safety requirements assignment rules prescribed by the target safety standard?

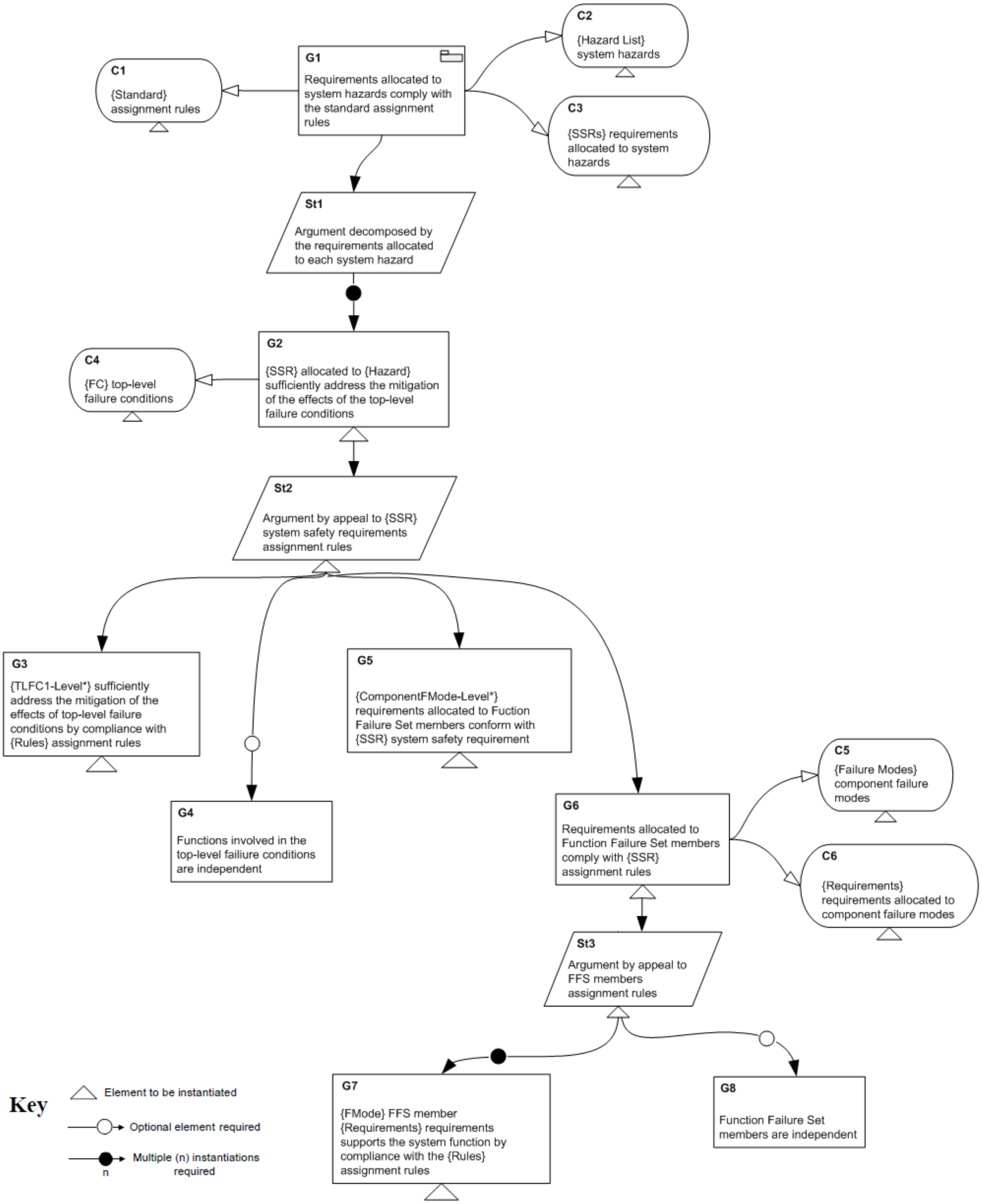
2.4 Forces

- Structuring a body of evidence arguing the compliance of system-level and component-level requirements allocation with the safety standard assignment rules can be complex. Documenting this argument in a structured argumentation style may reduce such complexity and providing a reusable solution for arguing the compliance of the requirements allocation with the rules established in the target safety standard;
- Verifying the compliance of the requirements allocation provided by model-based safety analysis tools with the assignment rules prescribed by the target safety standard may contribute to build an evidence item that can be used for certification proposals.

2.5 Solution

An important aspect of this pattern is that it divides and conquers the goal of compliance of the requirements allocated to system hazards (G1) with the assignment rules (C1) according to the stringency of the requirements (C3) allocated to each system hazard (C2). Details of the GSN notation elements used to describe the pattern can be found in the appendix.

2.6 Structure



G1
Requirements allocated to system hazards comply with the standard assignment rules
C1
{Standard} assignment rules
C2
{Hazard List} system hazards
C3
{SSRs} requirements allocated to system hazards
St1
Argument decomposed by the requirements allocated to each system hazard
G2
{SSR} allocated to {Hazard} sufficiently address the mitigation of the effects of the top-level failure conditions
C4
{FC} top-level failure conditions
St2
Argument by appeal to {SSR} system safety requirements assignment rules
G3
{TLFC1-Level*} sufficiently address the mitigation of the effects of top-level failure conditions by compliance with {Rules} assignment rules
G4
Functions involved in the top-level faiilure conditions are independent
G5
{ComponentFMode-Level*} requirements allocated to Fuction Failure Set members conform with {SSR} system safety requirement
G6
Requirements allocated to Function Failure Set members comply with {SSR} assignment rules
C5
{Failure Modes} component failure modes
C6
{Requirements} requirements allocated to component failure modes
St3
Argument by appeal to FFS members assignment rules
G7
{FMode} FFS member {Requirements} requirements supports the system function by compliance with the {Rules} assignment rules
G8
Function Failure Set members are independent
Key
Element to be instantiated
Optional element required
Multiple (n) instantiations required
n

Figure 1. System safety requirements decomposition compliance pattern structure.

2.7 Participants

G1: Defines the overall objective of the pattern.

St1: Decomposes G1 into sub-goals arguing the compliance of the requirements allocated to each system hazard with the assignment rules of the target safety standard.

G2: A sub-goal arguing the compliance of the requirements allocated to a particular hazard with the system-level assignment rules. A safety-critical system can be associated with multiple system-level requirements allocation that should be argued regarding the compliance with the assignment rules.

St2: Decomposes G2 into a set of system-level (G3, G4, and G5 sub-goals) and component-level assignment rules (G6 sub-goal).

G3: Argues the compliance of the requirements allocated to system functions that contribute the occurrence of a particular hazard with the system-level (i.e., function) assignment rules.

G4: An optional goal arguing the system functions contributing to the occurrence of a particular hazard are independent. Functional independence is an attribute where the system Functions are different in order to minimize the likelihood of a common requirement error. For example two different sets of functional requirements could minimize the likelihood of the same error.

G5: Argues the requirements allocated to each component failure conform with the requirements allocated to a particular system hazard.

G6: Argues the compliance of the requirements allocated to Function Failure Set (FFS) members (i.e., component-level failures) contributing to the occurrence of a particular hazard with component-level (i.e., item) assignment rules.

St3: Decomposes G6 into sub-goals arguing the compliance of the requirements allocated to FFS members (i.e., component failures) with item-level assignment rules; and an optional sub-goal arguing the independence of the components (i.e., items) involved in the hazard fault tree.

G7: Argues the requirements allocated to an individual FFS member (i.e., a component-level failure) comply with item-level (i.e., component) assignment rules. A FFS may include multiple members requiring multiple instances of G7.

G8: An optional goal arguing that the items (i.e., components) involved in the FFS are independent. Item independence is an attribute where the items are different to minimize the likelihood of a common mode error between the individually developed items.

C1: A context identifying the assignment rules underlying the target safety standard.

C2: A context identifying the system hazards.

C3: A context identifying the safety requirements allocated to the system hazards.

C4: The definition of the top-level failure conditions associated to a particular system hazard.

C5: A context identifying the component failure modes contributing to a particular system hazard.

C6: A context identifying the requirements allocated to each component failure mode.

2.8 Applicability

This safety case pattern can be used as a backing argument for Risk (Habli, 2009) and Hazard Mitigation (Kelly and McDermid, 1997) arguments, and Component Contributions to System Hazards pattern catalogue (Weaver, 2003), by substantiating the mitigation measures (i.e., the safety requirements) applied to minimize the effects of hazards and their associated component failure modes.

The proposed pattern is applicable in an automated verification scenario to generate an argument over the compliance of system safety requirements decomposition provided by model-based safety analysis tools, like HiP-HOPS, with the assignment rules of the target safety standard. HiP-HOPS provides the automatic generation of fault trees, FMEA, and requirements allocation from a system model annotated with hazard analysis and component failure information (Azevedo et al. 2014).

The pattern requires the following information to be instantiated:

- The requirements assignment rules underlying the target safety standard (i.e.C1 in Figure 1) (See participants section);
- The top-level failure conditions (C2) and the low-level failures that can cause the hazard. Fault tree analysis provides such information (See participants section);
- The requirements allocated to system hazards and their decomposition throughout the component failures involved in the fault tree (G2, G5, G6, and G7). Model-based safety analysis provides the requirements allocation information.

2.9 Collaborations

The argumentation over compliance of the requirements allocated to each particular system hazard (G2) is in the context of the top-level failure conditions which can cause the hazard (C2). G2 is decomposed into a set of system-level (G3, G4, and G5) and component-level (G6) assignment rules. These rules may change according to the stringency of the system safety requirements stated in G2, and whether the system functions are independent or not.

G6 is substantiated by arguing that the requirements allocated to each component-level failure (C5, and C6) comply with item development assignment rules (G7), and by an optional argument arguing the item (i.e., component) development independence (G8). The assignment rules specified in G7 may also change according to the stringency of the system safety requirements and whether the components are independent or not.

2.10 Consequences

- After applying this pattern, the evidence of the system safety requirements and requirements allocated to component failures compliance with functional and item requirements assignment rules is obtained;
- The application of this pattern may also contribute to verify the compliance of requirements allocation provided by model-based safety analysis tools with the assignment rules of the target safety standard;
- Instantiations of this argument pattern do not contain references to undeveloped/uninstantiated claims and arguments that require further development;
- Applying the pattern manually for a large system can be complex, requiring automation. Thus, the design of a mechanism to automatically generate pattern instances from the information provided by safety analysis assets (e.g., fault trees, FMEA, requirements allocation) is required.

2.11 Implementation

The implementation of this pattern firstly involves instantiating the contexts C1, C2, and C3. C1 provides the reference for the target safety standard and its functional and item requirements assignment rules. In the context of these rules, the requirements allocated to each system hazard and associated component failures are verified regarding the compliance with the assignment rules (G3, G6, and G7). C2 and C3 provide the information required to argue the compliance of the requirements allocated to the hazard top-level failure conditions with functional requirements assignment rules (G2).

For each requirement allocated to a system hazard, an argument must be constructed (by instantiating G2, G3, and G5) to demonstrate the compliance of the requirements allocated to system functions with functional assignment rules. As we can have multiple instances of this argument (i.e., G2), they can be represented as argument modules in an instance of the pattern. C4 must also be instantiated to provide the top-level failure conditions (i.e., failures in the system functions) involved in a particular system hazard. The instantiation of G3 requires the safety requirements allocated to each top-level failure condition and functional requirements assignment rules. These rules change according to the stringency of the requirements allocated to the system hazard (i.e., SSR referenced in St2 strategy node).

G4 is applicable to safety standards (e.g., ARP 4754A) where functional independence attribute may change the assignment rules. G4 is included in the pattern instance if the system functions are independent. The analysis of the failure expressions of a hazard (i.e., combination of system failures which lead directly to the hazard) provides the information about the independence of the system functions. The system functions are independent when the failure expressions are connected via AND gates, meaning all system functions must fail to cause the hazard. The system functions are not independent when the failure expressions are connected via OR gates. It means a failure in one of these functions is sufficient to cause the hazard.

The instantiation of G5 requires the definition of the requirements allocated to each component failure and the system safety requirement. An argument must also be constructed (by instantiating G6 and G7) to demonstrate the requirements allocated to each component failure (i.e., a Function Failure Set member) comply with item requirements assignment rules. C5 and C6 provide the definition of the component failures and their associated requirements required to instantiate G7. The instantiation of G7 also requires the system safety requirement allocated to the hazard. An instance of G7 must exist for each component failure (i.e., a Function Failure Set member) contributing to the hazard.

G8 is applicable to safety standards (e.g., ARP 4754A) where item development independence may change the assignment rules. G8 is included in the pattern instance if the components involved in a system hazard are independent. The analysis of the functional failure set (i.e., it describes how component failures are connected in a fault tree) provides the information about the item (component) development independence. The components are independent when the component failures are connected via AND gates, meaning all component failures of tree must fail to cause the hazard. The components are not independent when component failures are connected via OR gates. It means the occurrence of a component failure is sufficient to cause the hazard.

Possible pitfalls:

- Not providing the reference for the target safety standard (C1);
- Incomplete definition of the system hazards and their associated safety requirements (C2 and C3);
- Not providing all the top-level failure conditions of a system hazard (C4);
- Imprecise information about the component failures of a hazard and their associated safety requirements (C5 and C6).

Qualified model-based safety analysis tools can be used for minimizing these pitfalls.

2.12 Example

The pattern was applied to a product of an automotive Hybrid Braking System software product line (HBS-SPL) developed based on ISO 26262; and the flight control system of an unmanned aircraft developed based on ARP 4754A. ISO 26262 and ARP 4754A provide different assignment rules for allocation and decomposition of system safety requirements throughout the failures in components of the system architecture. Safety requirements in ISO 26262 are called Automotive Safety Integrity Levels (ASILs) and in ARP 4754A they are called Development Assurance Levels (DALs).

ISO 26262 defines a simple integer algebra for ASIL decomposition. Each ASIL is equivalent to a number: QM (Quality Management) = 0, A = 1, B = 2, C= 3, and D = 4 (i.e., D is more stringent). ISO 26262 provides ASIL decomposition rules allowing redundant elements to share the burden of achieving a given ASIL. So, two components sharing responsibility for meeting ASIL D might individually only be required to meet ASIL B because it produces the same ASIL value (2 + 2 = 4). Table 1 presents these rules for each ASIL class. ISO 26262 ASIL assignment rules do not include assumptions about independence.

Table 1 – ISO 26262 ASIL decomposition rules (ISO, 2011).

ASIL Class	**ASIL Decomposition Rules**		
ASIL D	ASIL C(D) + ASIL A(D)	ASIL B(D) + ASIL B(D)	ASIL D(D) + QM(D)
ASIL C	ASIL B(C) + ASIL A(C)	ASIL C(C) + QM(C)	
ASIL B	ASIL A(B) + ASIL A(B)	ASIL B(B) QM(B)	
ASIL A	ASIL A(A) + QM(A)		

Figure 2 presents the pattern instantiation considering the safety requirements allocated to system hazards and component failures of an HBS-SPL product obtained from safety analysis. ISO 26262 was defined as the target safety standard by instantiating C1. C2 and C3 were instantiated with the HBS-SPL product hazards and allocated ASILs. For each ASIL allocated to a system hazard, G2 (Figure 1) was instantiated to build an argument to demonstrate the compliance of ASILs allocated to a particular system hazard and their component failure modes with the assignment rules. For example, G3 was instantiated to argue the ASIL D allocated to “No braking four wheels” system hazard. G3 is in the context of the following top-level failure conditions: “Omission-Brake_Unit1.Add.Braking AND Omission-Brake_Unit2.Add.Braking AND Omission-Brake_Unit3.Add.Braking AND Omission-Brake_Unit4.Add.Braking” (C4). Members of the top-level failure conditions stand for failures in system functions.

In order to avoid complexity of the model, GSN modular extensions (Habli and Kelly, 2010) were used to place the arguments related to the remaining system-level ASILs in different argument modules. For example, G2_Module is a reference to the argument module that argues the compliance of the ASIL D allocated to “No braking three wheels” hazard with the assignment rules. G3 is decomposed into a set of ASIL D assignment rules by instantiating St2. G4 was instantiated to argue that the ASILs allocated to the top-level failure conditions of the “No braking four wheels” hazard comply with ASIL D assignment rules.

As ISO 26262 ASIL assignment rules do not include assumptions about functional independence, G4 (see Figure 1) was not included in the argumentation of G3. G5 was instantiated to argue that the ASILs allocated to each component failure mode contributing to the hazard conform with the ASIL allocated to the system hazard. Finally, G6 assignment rule was instantiated to argue that the ASILs allocated to each component failure mode comply with ASIL D assignment rules. C5 and C6 were instantiated to provide the component failure modes involved in “No braking four wheels” fault tree (i.e., function failure set members) and the ASILs allocated to these failure modes for G6.

G6 is decomposed into sub-goals arguing the compliance of ASILs allocated to each component failure mode with the ASIL D assignment rules. For each component failure mode, one instance of G7 (see Figure 1) was created to argue that the ASIL allocated to this component failure mode complies with ASIL D assignment rules. Thereby, we can have multiple instances of this goal. G7 and G8 argue that the ASILs allocated to “Omission-Brake_Unit1.EMB.Out1” and “Omission-Communication_ Bus1.Out1” failure modes comply with ASIL D assignment rules. G8 (see Figure 1) was not included in the argument because ISO 26262 ASIL assignment rules do not include assumptions about item development independence.

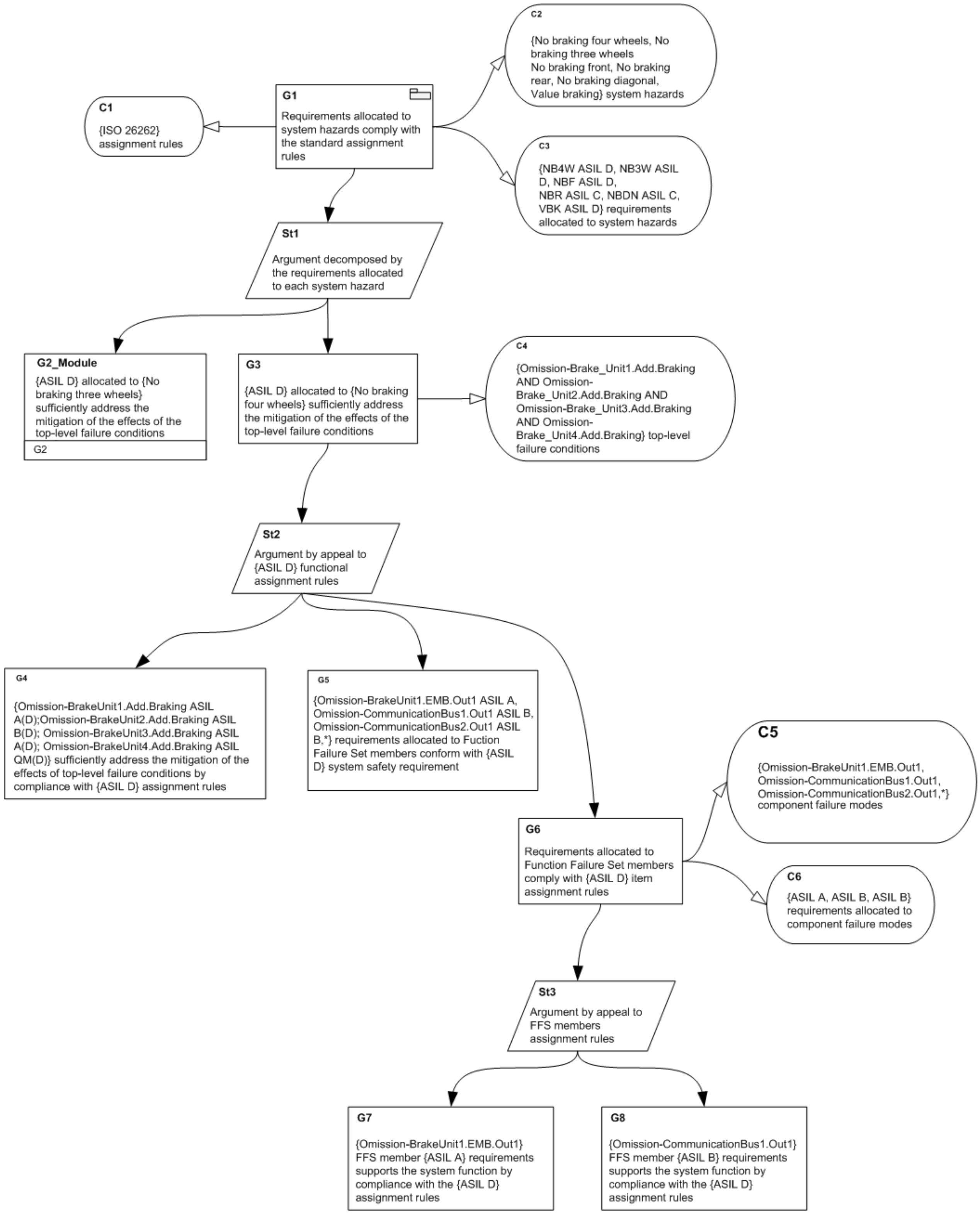

C2
{No braking four wheels, No braking three wheels
No braking front, No braking rear, No braking diagonal, Value braking} system hazards
C1
{ISO 26262} assignment rules
G1
Requirements allocated to system hazards comply with the standard assignment rules
C3
{NB4W ASIL D, NB3W ASIL D, NBF ASIL D,
NBR ASIL C, NBDN ASIL C, VBK ASIL D} requirements allocated to system hazards
St1
Argument decomposed by the requirements allocated to each system hazard
G2_Module
{ASIL D} allocated to {No braking three wheels} sufficiently address the mitigation of the effects of the top-level failure conditions
G2
G3
{ASIL D} allocated to {No braking four wheels} sufficiently address the mitigation of the effects of the top-level failure conditions
C4
{Omission-Brake_Unit1.Add.Braking AND Omission-Brake_Unit2.Add.Braking AND Omission-Brake_Unit3.Add.Braking AND Omission-Brake_Unit4.Add.Braking} top-level failure conditions
St2
Argument by appeal to {ASIL D} functional assignment rules
G4
{Omission-BrakeUnit1.Add.Braking ASIL A(D);Omission-BrakeUnit2.Add.Braking ASIL B(D); Omission-BrakeUnit3.Add.Braking ASIL A(D); Omission-BrakeUnit4.Add.Braking ASIL QM(D)} sufficiently address the mitigation of the effects of top-level failure conditions by compliance with {ASIL D} assignment rules
G5
{Omission-BrakeUnit1.EMB.Out1 ASIL A, Omission-CommunicationBus1.Out1 ASIL B, Omission-CommunicationBus2.Out1 ASIL B,*} requirements allocated to Fuction Failure Set members conform with {ASIL D} system safety requirement
C5
{Omission-BrakeUnit1.EMB.Out1, Omission-CommunicationBus1.Out1, Omission-CommunicationBus2.Out1,*} component failure modes
G6
Requirements allocated to Function Failure Set members comply with {ASIL D} item assignment rules
C6
{ASIL A, ASIL B, ASIL B} requirements allocated to component failure modes
St3
Argument by appeal to FFS members assignment rules
G7
{Omission-BrakeUnit1.EMB.Out1} FFS member {ASIL A} requirements supports the system function by compliance with the {ASIL D} assignment rules
G8
{Omission-CommunicationBus1.Out1} FFS member {ASIL B} requirements supports the system function by compliance with the {ASIL D} assignment rules

Figure 2. Pattern instance in an automotive system developed based on ISO 26262.

System Safety Requirements Compliance argument pattern was also used to argue the safety of an unmanned avionics flight control system developed based on ARP 4754A. ARP 4754A provides a set of

DAL assignment rules for system safety requirements (i.e., functional requirements in this standard), and component failure requirements (i.e., item development requirements). Table 2 presents an overview of the DAL assignment rules. These rules are applicable at both functional and item levels and also allow two or more FFS members sharing responsibility for meeting a more stringent DAL. For example, DAL B and DAL D allocated to two FFS members are sufficient to meet DAL B.

Table 2 – DAL assignment rules (EUROCAE, 2010).

Top-Level Failure Condition Classification	Functional Failure Sets with a Single Member	Functional Failure Sets with Multiple Members	
		Option 1	**Option 2**
Hazardous/Severe Major	FDAL B	FDAL B for one Member, and additional Member(s) with FDAL not lower than D.	FDAL C for two of the Members leading to the top-level Failure Condition. Additional Member(s) with FDAL not lower than D.
Major	FDAL C	FDAL C for one Member, and additional Member(s) with FDAL not lower than E.	FDAL D two Members leading to the top-level Failure Condition. Additional Members with FDAL E.
Minor	FDAL D	FDAL D for one Member, and additional Members with FDAL E.	
No safety effect	FDAL E	FDAL E	
Note 1: When FDAL A is allocated to a single FFS Member, the applicant may be required to substantiate that the development process for that Member has sufficient independent validation/verification activities, techniques and completion criteria to ensure the mitigation of catastrophic effects of potential development error (s); Note 2: For a catastrophic Failure Condition any degree of decomposition from a top FDAL A FFS should include at least one FDAL A or two FDAL B. This rule is valid for any row above.			

Figure 3 presents the pattern instantiation considering the safety requirements allocated to system hazards and component failures of an unmanned avionics flight control system obtained from safety analysis. ARP 4754A was defined as the target safety standard by instantiating C1. C2 and C3 were instantiated with the system hazards and allocated DALs. For each DAL allocated to a particular system hazard, G2 (Figure 1) was instantiated to build an argument to demonstrate the compliance of DALs allocated hazards and their component failure modes with the assignment rules. In this case, G3 was instantiated to argue the DAL C allocated to “Value velocity” system hazard. G3 is in the context of the following top-level failure conditions: “Value-ThreeAxisGyroscope.out1 AND Value-GPS.bus1.Out1” (C4).

GSN modular extensions were also used here to reduce the complexity of the model by placing the arguments related to the remaining DALs in different argument modules. For example, G2_Module is a reference for the argument module which argues the compliance of the DAL B allocated to “”Value altitude” hazard with the assignment rules. G3 is decomposed into a set of DAL C assignment rules by instantiating St2. G4 is instantiated to argue that the DALs allocated to the top-level failure conditions of “Value velocity” hazard comply with DAL C assignment rules.

As the top-level failure conditions are connected via “AND” gates in the hazard expression (see C4), G5 was included in the argumentation of G3. G6 was instantiated to argue that the DALs allocated to each component failure mode contributing to the hazard conform with the DAL allocated to the system hazard. G7 assignment rule was instantiated to argue that the DALs allocated to each component failure mode comply with DAL C assignment rules. C5 and C6 were instantiated to provide the component failure modes involved in “Value velocity” fault tree (i.e., functional failure set members) and the DALs allocated to these failure modes for G7.

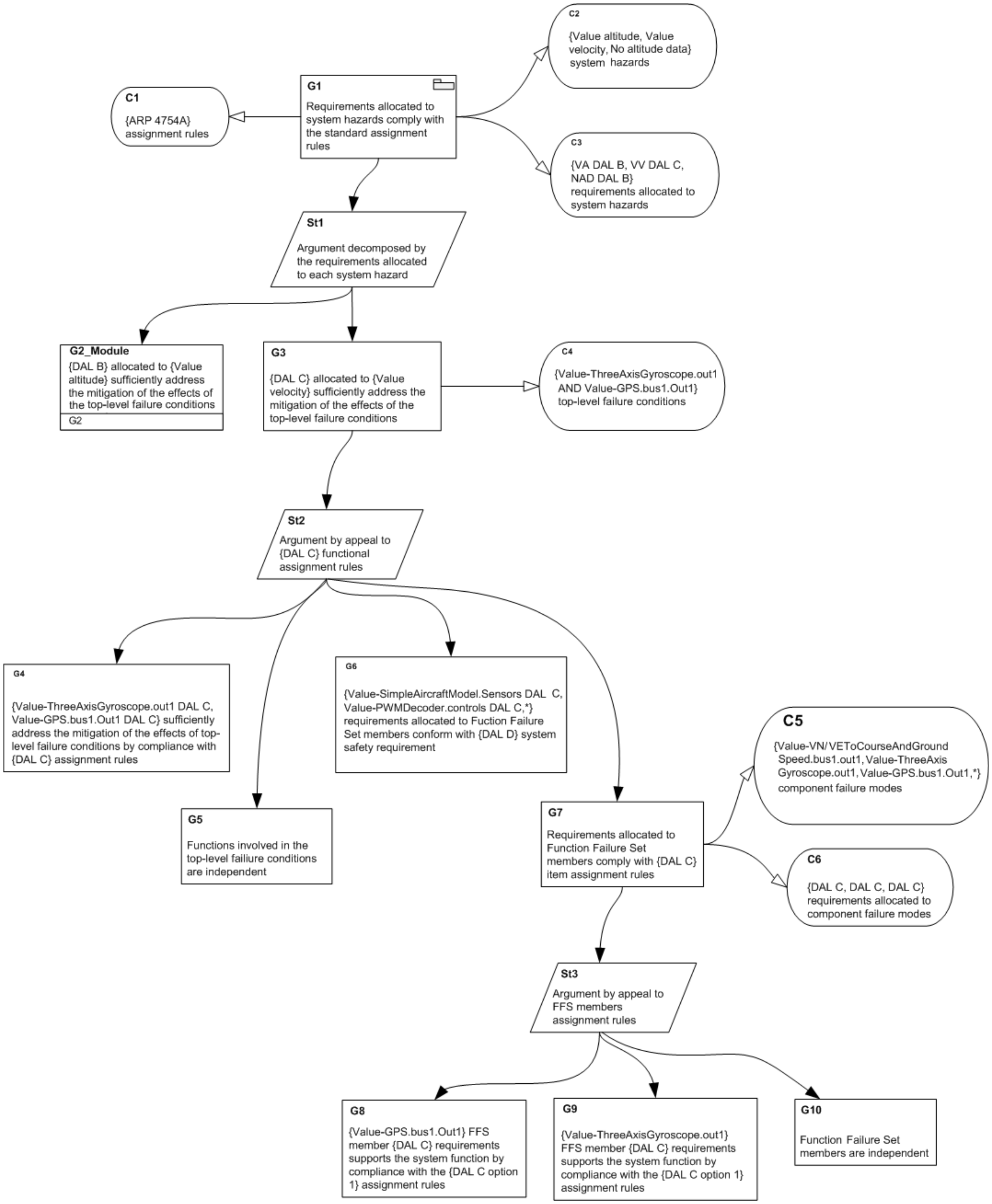

C2
{Value altitude, Value velocity, No altitude data} system hazards
C1
{ARP 4754A} assignment rules
G1
Requirements allocated to system hazards comply with the standard assignment rules
C3
{VA DAL B, VV DAL C, NAD DAL B} requirements allocated to system hazards
St1
Argument decomposed by the requirements allocated to each system hazard
G2_Module
{DAL B} allocated to {Value altitude} sufficiently address the mitigation of the effects of the top-level failure conditions
G2
G3
{DAL C} allocated to {Value velocity} sufficiently address the mitigation of the effects of the top-level failure conditions
C4
{Value-ThreeAxisGyroscope.out1 AND Value-GPS.bus1.Out1} top-level failure conditions
St2
Argument by appeal to {DAL C} functional assignment rules
G4
{Value-ThreeAxisGyroscope.out1 DAL C, Value-GPS.bus1.Out1 DAL C} sufficiently address the mitigation of the effects of top-level failure conditions by compliance with {DAL C} assignment rules
G6
{Value-SimpleAircraftModel.Sensors DAL C, Value-PWMDecoder.controls DAL C,*} requirements allocated to Fuction Failure Set members conform with {DAL D} system safety requirement
C5
{Value-VN/VEToCourseAndGround Speed.bus1.out1, Value-ThreeAxis Gyroscope.out1, Value-GPS.bus1.Out1,*} component failure modes
G5
Functions involved in the top-level failiure conditions are independent
G7
Requirements allocated to Function Failure Set members comply with {DAL C} item assignment rules
C6
{DAL C, DAL C, DAL C} requirements allocated to component failure modes
St3
Argument by appeal to FFS members assignment rules
G8
{Value-GPS.bus1.Out1} FFS member {DAL C} requirements supports the system function by compliance with the {DAL C option 1} assignment rules
G9
{Value-ThreeAxisGyroscope.out1} FFS member {DAL C} requirements supports the system function by compliance with the {DAL C option 1} assignment rules
G10
Function Failure Set members are independent

Figure 3. Pattern instance in an unmanned avionics system developed based on ARP 4754A.

G7 is decomposed into sub-goals arguing the compliance of DALs allocated to each component failure mode with the DAL C assignment rules; and arguing the independence of these components (i.e., if these components were designed to minimize a common mode error). For each component failure mode, one instance of G7 (see Figure 1) was created to argue that the DAL allocated to this component failure mode complies with ASIL D assignment rules. G8 and G9 argue that the DALs allocated to "Value-GPS.bus1.Out1" and "Value-ThreeAxisGyroscope.Out1" failure modes comply with DAL C option 1 assignment rules. As the component failure modes are connected via "AND" gates in "Value velocity" fault tree, we conclude that these components are independent (G10).

2.13 Known Uses

This pattern was used to verify compliance of allocation of functional and item requirements in a Hybrid Braking System automotive system developed based on ISO 26262 (ISO, 2011), and Tiriba flight control unmanned avionics system developed based on ARP 4754A (EUROCAE, 2010). Besides being applied to automotive and avionics systems (as shown in Section 2.10), this pattern is also applicable to industry systems developed according to IEC 61508 (International Electrotechnical Commission, 2009).

2.14 Related Patterns

Risk Argument (Habli, 2009), Hazard Mitigation Argument (Kelly and McDermid, 1997), and Component Contributions to System Hazards pattern catalogue (Weaver, 2003).

APPENDIX: AN OVERVIEW OF THE GOAL STRUCTURING NOTATION (GSN)

Introduction

A Safety Case aims to provide a reasoned and compelling argument, supported by a body of evidence, that a system, service, or organization will operate as intended for a defined application in a defined environment (GSN Standard, 2011). In the sense used in safety cases (i.e. assurance cases) an argument is used to demonstrate how someone can reasonably conclude that a system is acceptably safe from the evidence available. An argument is defined as a connected set of claims intended to establish an overall claim. Some of these claims may need further support giving rise to a hierarch of claims by which an argument is established.

Goal Structuring Notation (GSN) is a graphical argument notation that can be used to explicitly represent the individual elements of an argument (claims, evidence, and context), and the relationships that exist between these elements (i.e. how individual claims are supported by specific claims, and how claims are supported by the evidence and the context defined for the argument) in a logical structure known as "goal structure". This appendix presents the following core GSN elements: Goal, Strategy, Solution, and Context; and GSN patterns extensions used to document the proposed "System Safety Requirements Decomposition Compliance" Safety Case pattern.

GSN Core

The main GSN notation elements are shown in Figure 4 with example instances of each concept. The claims of an argument are documented as GSN *goals* and the items of evidence as *solutions*. GSN establishes the following relationships between elements: a predicate-conclusion relationship between *goals* and *sub-goals*, and the support that *solutions* provide for *goals* denoted by "*Supported By*" relationships; and the relationship between GSN goal/or strategy and the context in which it is stated denoted by *"In Context Of"*.

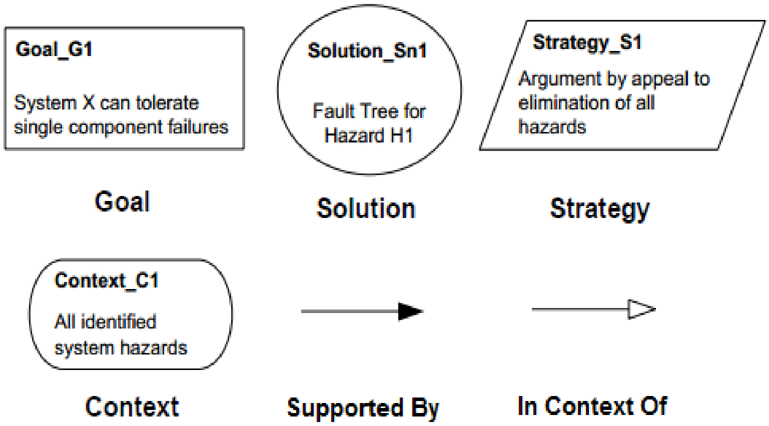

Goal_G1
System X can tolerate single component failures
Solution_Sn1
Fault Tree for Hazard H1
Strategy_S1
Argument by appeal to elimination of all hazards
Goal
Solution
Strategy
Context_C1
All identified system hazards
Context
Supported By
In Context Of

Figure 4. GSN core graphical elements (GSN Standard, 2011).

The following provides the conceptual definition of the GSN core elements presented in Figure 4:

Goal: is represented by a rendered rectangle. It is used to present claims and sub-claims that forms the argument. *Goal_G1* is example of a GSN goal with the statement of a claim arguing the system is tolerable to single component failures.

Strategy: is used to describe the nature of the inference to decompose a claim stated a *goal* into sub-claims stated in *sub-goals* denoted by a parallelogram. For example, the Strategy_S1 provide the inference "elimination of all hazards" to decompose a top-level goal stating the system is safe into sub-goals arguing the elimination of each system hazard.

Context: presents a contextual artefact which can be a reference to contextual information, or a statement. It is used to capture the context in which the claim (i.e. Goal) or reasoning step (i.e. Strategy) should be interpreted. In GSN a *Context* is rendered as a rounded rectangle. Figure 4 shows the Context_C1 which can be used to describe that the Strategy_S1 is stated in the context of "all identified system hazards".

Solution: is used to present a reference to an evidence item required to support the truth of a claim. A Solution is rendered as a circle. Figure 4 presents an example in which the *Solution_Sn1* asserts that the claim over the mitigation of "Hazard 1" is supported by a fault tree evidence item.

Supported By relationship: is represented by a line with a solid arrowhead, used for indicating inferential or evidential relationships between GSN elements. An inferential relationship states that there is an inference between goals in the argument. An evidential relationship declares the link between a goal and an evidence item used to substantiate it. Permitted "*Supported By"* connections according to GSN Standard (2011) are: goal-to-goal, goal-to-strategy, goal-to-solution, and strategy-to-goal.

In Context Of relationship: is represented by a line with a hollow arrowhead, used for declaring contextual relationships between GSN Goal/Strategy and Context elements. Permitted "*In Context Of"* connections according to GSN Standard (2011) are: goal-to-context, strategy-to-context; and goal-to-assumption, goal-to-justification, strategy-to-assumption, and strategy-to-justification connections not used in the pattern description.

Figure 5 illustrates an example of goal structure with a top-level goal arguing the "C/S (Control System) logic is fault free". This claim is further broken down into sub-goals indirectly through two argument strategies putting forward "Argument by satisfaction of all C/S safety requirements", and "Argument by omission of all identified software hazards" as a mean of addressing the top-level goal (i.e. G1). These strategies are substantiated by G2, G3, G4, G8, and G9 sub-goals.

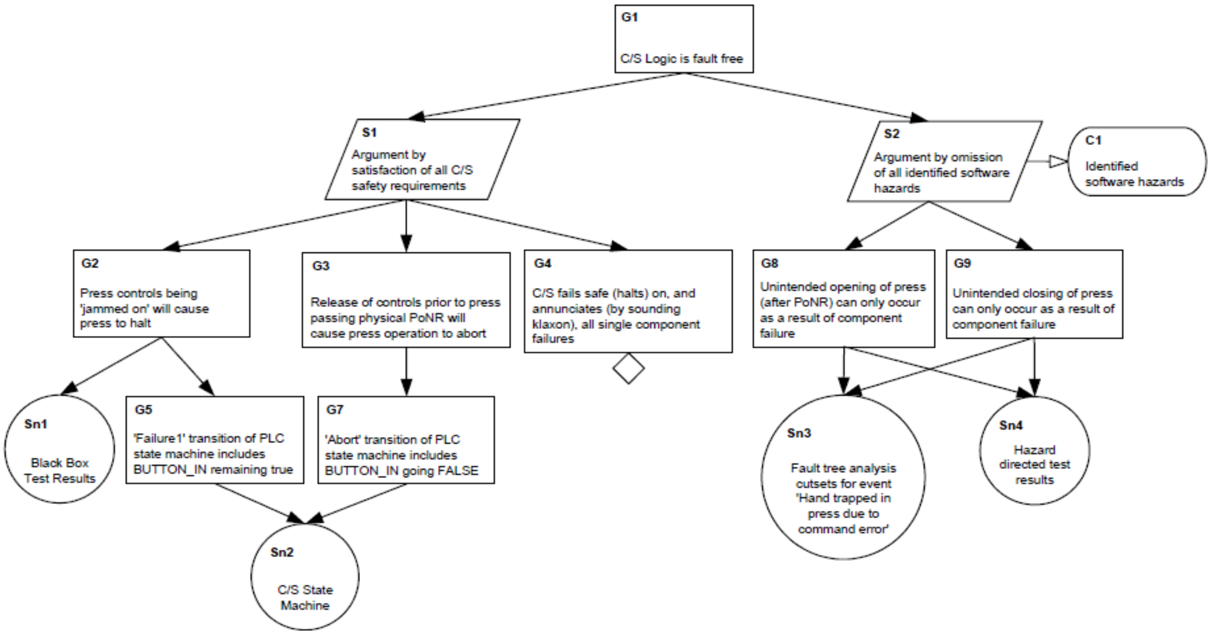

G1
C/S Logic is fault free
S1
Argument by satisfaction of all C/S safety requirements
S2
Argument by omission of all identified software hazards
C1
Identified software hazards
G2
Press controls being 'jammed on' will cause press to halt
G3
Release of controls prior to press passing physical PoNR will cause press operation to abort
G4
C/S fails safe (halts) on, and annunciates (by sounding klaxon), all single component failures
G8
Unintended opening of press (after PoNR) can only occur as a result of component failure
G9
Unintended closing of press can only occur as a result of component failure
Sn1
Black Box Test Results
G5
'Failure1' transition of PLC state machine includes BUTTON_IN remaining true
G7
'Abort' transition of PLC state machine includes BUTTON_IN going FALSE
Sn3
Fault tree analysis cutsets for event 'Hand trapped in press due to command error'
Sn4
Hazard directed test results
Sn2
C/S State Machine

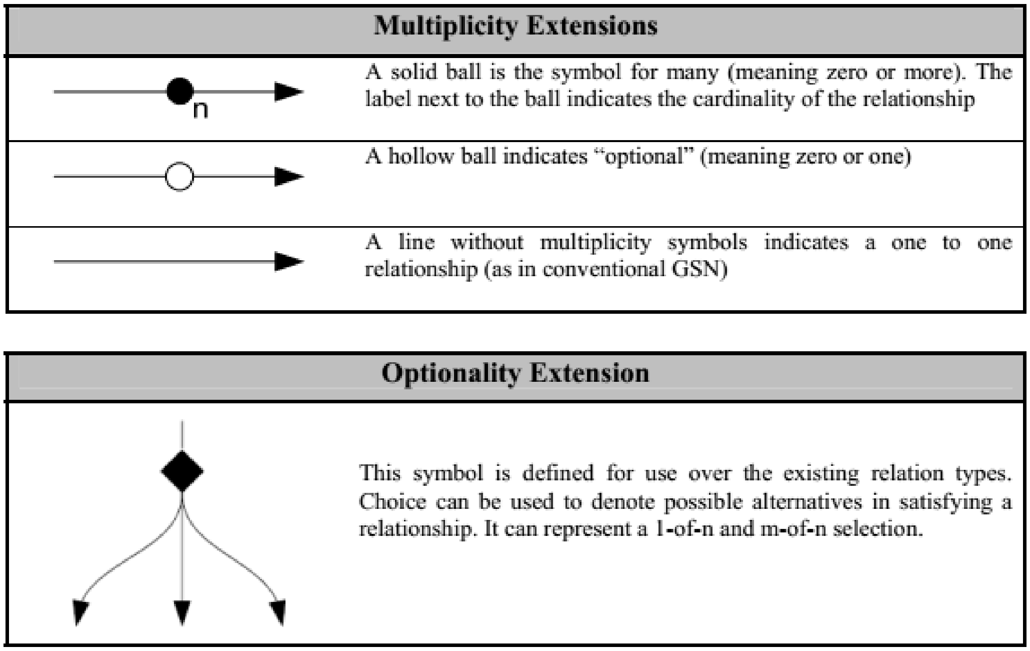
Figure 5. An example of GSN goal structure (Kelly and Weaver, 2004).

A goal statement like “Unintended closing of press after PoNR (Point of No Return) can only occur as a result of component failure” in G8, is supported by direct references to solutions: “Fault Tree cutsets” and “Hazard Directed Testing Results”. Otherwise, G2, G3, and G4 goal statements are supported by other sub-goals or safety arguments.

GSN Patterns Extension

Similar to GoF object-oriented software design patterns (Gamma et al. 1995), safety case patterns are created in GSN to abstract the details of recurrent safety argument structures. Safety case patterns describe successful and proven style of argumentation rather than a concrete argument for a particular. In order to address safety case patterns, the GSN was extended by adding:

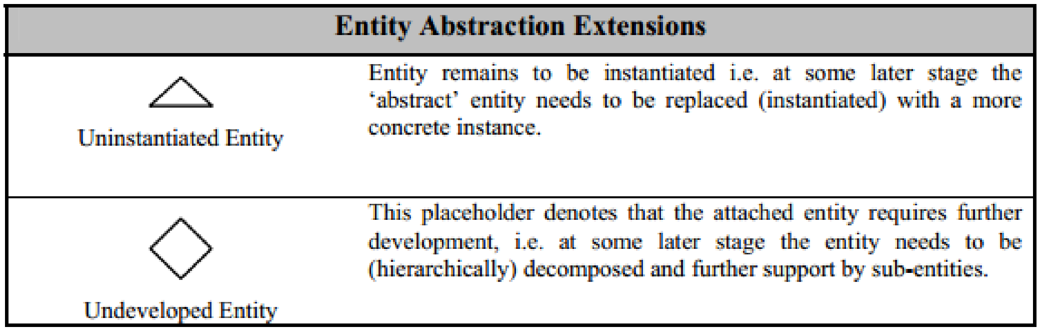
- **Structural Abstractions**: supporting generalized n-ary, optional, and multiplicity alternative relationships between GSN elements as presented in Figure 6; and
- **Entity Abstractions**: supporting generalization/specialization of GSN elements (Figure 7).



Multiplicity Extensions	
──●→ n	A solid ball is the symbol for many (meaning zero or more). The label next to the ball indicates the cardinality of the relationship
──○→	A hollow ball indicates “optional” (meaning zero or one)
──→	A line without multiplicity symbols indicates a one to one relationship (as in conventional GSN)

Optionality Extension	
◆	This symbol is defined for use over the existing relation types. Choice can be used to denote possible alternatives in satisfying a relationship. It can represent a 1-of-n and m-of-n selection.

Figure 6. GSN structural abstractions (Habli and Kelly, 2010).



Entity Abstraction Extensions	
△ Uninstantiated Entity	Entity remains to be instantiated i.e. at some later stage the ‘abstract’ entity needs to be replaced (instantiated) with a more concrete instance.
◇ Undeveloped Entity	This placeholder denotes that the attached entity requires further development, i.e. at some later stage the entity needs to be (hierarchically) decomposed and further support by sub-entities.

Figure 7. GSN entity abstractions (Habli and Kelly, 2010).

Figure 8 illustrates how the GSN pattern notation can be used to represent the “Diverse Argument” domain-independent argument pattern. This pattern provides an argumentation style which is robust against single points of failures, i.e., sub-goals (Gn, G2) independently supporting the top-level goal. G1 is a parameterized (i.e. denoted by “{ }”) goal that should be instantiated. S1 strategy is connected to an optional and uninstantiated context (C1). This strategy is substantiated by an optional undeveloped sub-goal (i.e. G2) and by multiple occurrences of Gn sub-goal.

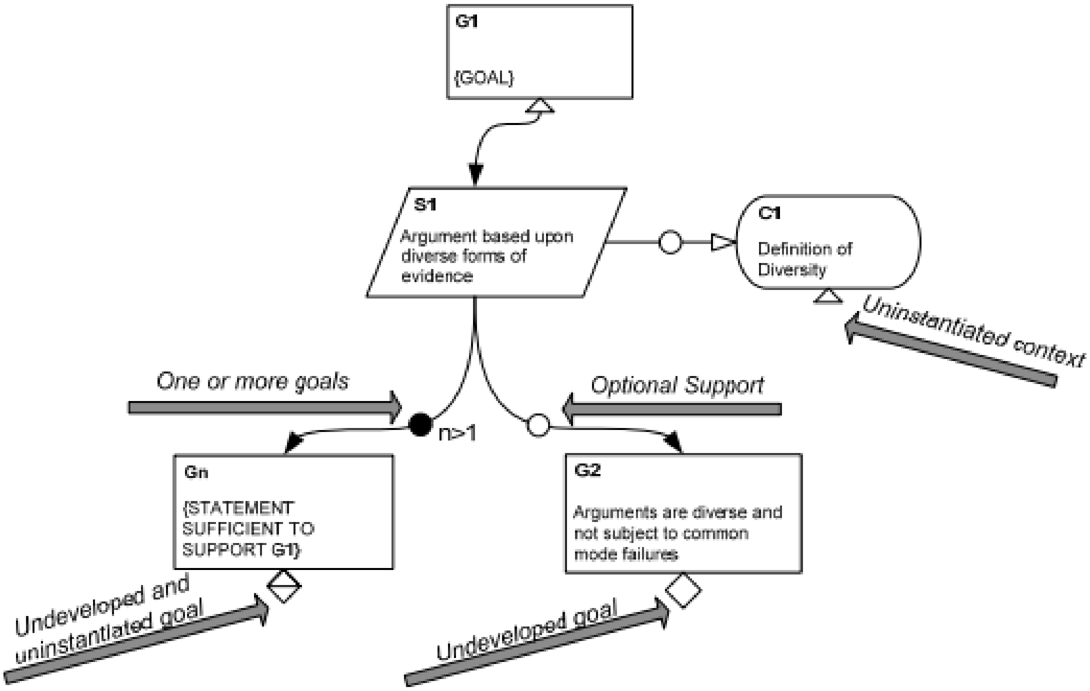

G1
{GOAL}
S1
Argument based upon diverse forms of evidence
C1
Definition of Diversity
Uninstantiated context
One or more goals
Optional Support
n>1
Gn
{STATEMENT SUFFICIENT TO SUPPORT G1}
G2
Arguments are diverse and not subject to common mode failures
Undeveloped and uninstantiated goal
Undeveloped goal

Figure 8. Pattern description using GSN extensions (Habli, 2009).

REFERENCES

Azevedo, L, Parker, D., Walker, M., Papadopoulos, Y., Araujo, R., 2014. Assisted Assignment of Automotive Safety Requirements. IEEE Software 31(1):62-68.

EUROCAE, 2010. ARP4754A/ ED-79A - Guidelines for Development of Civil Aircraft and Systems.

Gamma, E., Helm, R., Johnson, R. and Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software, Addison Wesley, 1995.

GSN Standard, 2011. GSN community standard version 1. Available on-line: http://www.goalstructuring notation.info/documents/GSN_Standard.pdf

Habli, I., and Kelly, T., 2010. A safety case approach to assuring configurable architectures of safety-critical product lines", Proceedings of the 1st International Conference on Architecting Critical Systems, Springer-Verlag, pp. 142-160.

Habli, I., Model-based Assurance of Safety-Critical Product Lines, Ph.D thesis, Department of Computer Science, University of York, 2009.

International Electrotechnical Commission, 2009. IEC 61508, Functional Safety of Electrical/ Electronic/Programmable Electronic Safety Related Systems - Part 3: Software requirements, 65A/550/FDIS, IEC.

International Organization for Standardization, 2011. ISO 26262: Road Vehicles Functional Safety.

Kelly, T. and Weaver, R, 2004. The goal structuring notation -a safety argument notation. In Proceedings of the Dependable Systems and Networks, Workshop on Assurance Cases.

Kelly, T. 2003. A systematic approach to safety case management", SAE world congress, Society for Automotive Engineers.

Kelly, T. and McDermid, J. 1997. Safety Case Construction and Reuse Using Patterns. Proceedings of the 16th International Conference on Computer Safety, Reliability and Security, pp 55-69.

Weaver, R. A. The Safety of Software Constructing and Assuring Arguments, Ph.D thesis, Department of Computer Science, University of York, 2003.