SAFETY CASE PATTERNS – REUSING SUCCESSFUL ARGUMENTS

Tim Kelly, John McDermid

Rolls-Royce Systems and Software Engineering
University Technology Centre
Department of Computer Science
University of York
Heslington
York YO10 5DD

Tel: (01904) 434728
Fax: (01904) 432708

E-mail: tim.kelly@cs.york.ac.uk

1. Context

The purpose of a safety case is to argue that a system is acceptably safe to operate in a specified context This *argument* demonstrates how the available *evidence* can be interpreted as compliance with the applicable *safety objectives.* As such, one might expect each safety case to be a unique synthesis of the *particular* pieces of evidence, safety objectives etc. However, we have observed common, re-used, approaches to arguing safety (especially within well-defined domains) – repeated structures of 'successful' (i.e. correct, comprehensive and convincing) arguments.

Informal reuse of safety case arguments is already commonplace – i.e. using 'largely the same' arguments of safety as used on previous projects. This form of reuse often occurs through 'Cut and Paste' of the textual safety case documents between projects. However, there are a number of problems with such an approach:

- It can be difficult to identify opportunities for reuse (i.e. take full advantage of successful arguments)
- Reuse occurs in an ad-hoc fashion – in a way that cannot be predicted or depended upon for project management
- Inappropriate reuse occurs. The context of a safety argument may not be exactly the same from one instance to another. Critical assumptions may be challenged.
- Lack of traceability. There is difficulty in knowing where arguments have been repeated. Problems can arise if 'faulty' arguments are propagated.
- Lack of consistency / process maturity – different (sometimes only subtly different) argument approaches may be unnecessarily used where reuse would improve consistency of approach and better support claims of a mature process.
- Loss of knowledge. There is no mechanism or medium for recording the essential 'best practice' of safety case development / safety argument construction.

For the nature and level of reuse we anticipated the approach of identifying and describing 'patterns' of safety argument seem to fit naturally with our objectives:

- We do not wish to define *whole* reusable safety cases – just elements, themes and structuring concepts within the safety case.
- We are not interested in establishing *hard and fast* rules to be applied in safety case construction. Instead, we wish to provide guidance and exemplars that can be adapted according to individual situations.

- We need to ensure that the reusable arguments were well documented with concepts such as intent, context and applicability captured (to avoid the problem of inappropriate reuse).

Having a clear representation of safety arguments was a pre-requisite to attempting any form of safety argument reuse. (It is extremely difficult to recognise and capture a reusable concept if there is no means of describing it!). The Goal Structuring Notation (GSN) [Wilson95] is a graphical notation developed in York explicitly for the purpose of representing safety case arguments. The next section provides a brief overview of the notation.

2. The Goal Structuring Notation

The principal elements of the notation are shown in Figure 1(with example instances of each concept).

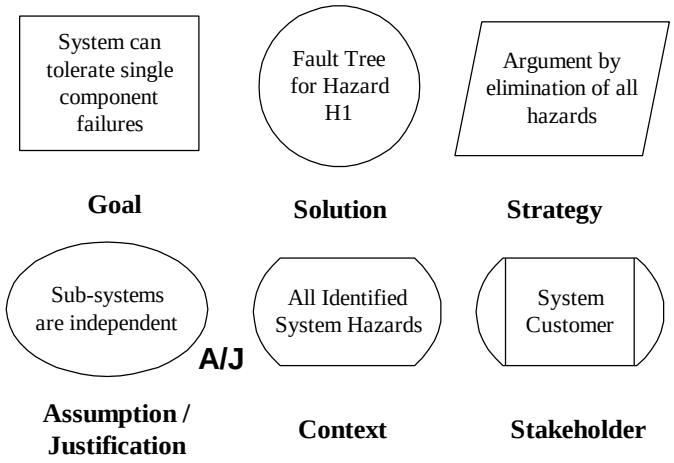


Figure 1 - Principal Elements of the Goal Structuring Notation

These elements are placed together to form a goal structure. The purpose of a goal structure is to show how **goals** are broken down into sub-goals, and eventually supported by evidence (**solutions**) whilst making clear the **strategies** adopted, the rationale for the approach (**assumptions, justifications**) and the **context** in which goals are stated. For further details on GSN see [Wilson95].

Figure 2 shows an example goal structure with some of the basic elements placed together to form a safety argument (albeit a heavily simplified one in this case).

> *In this figure, the argument that the control system is safe is based on two fundamental claims – firstly one of hazard avoidance, and secondly one of appropriate development processes. The context of these two claims is set out clearly (i.e. the identified hazards, the definition of acceptable mitigation and the 'acceptable' development process guidelines). These claims are then broken down further to address particular hazards and particular system elements. Justification is given for the particular failure rate target stated. Eventually, the forms of evidence used to support these basic claims are stated.*

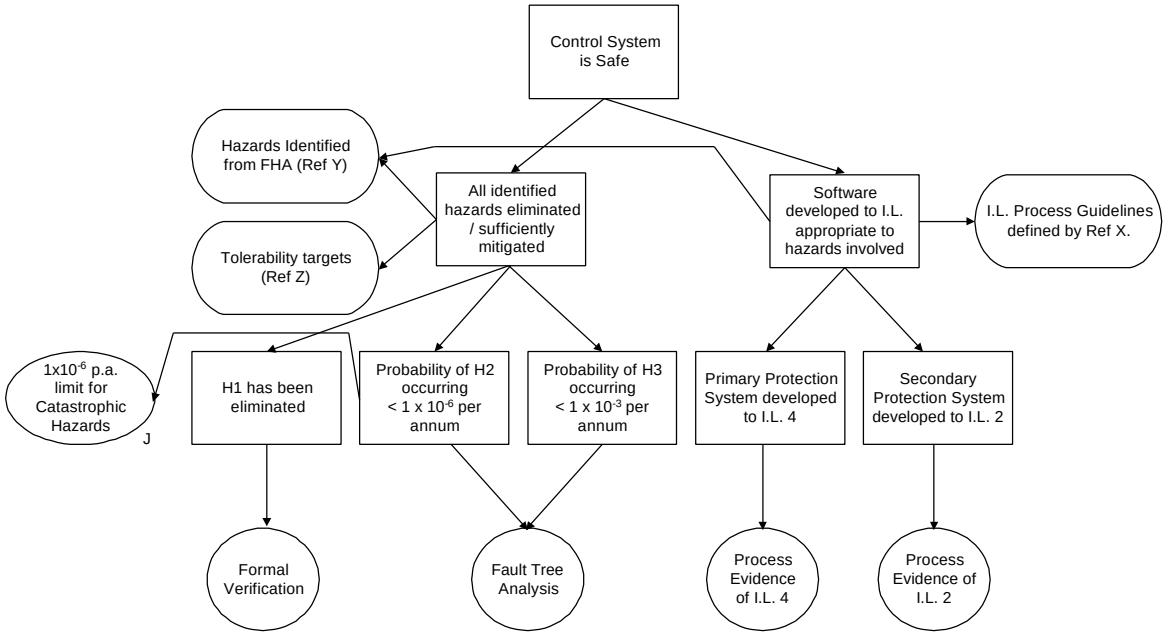


Figure 2 - An Example Goal Structure

3. Patterns

The concept of Design and Analysis Patterns in software design is receiving increasing interest and support. (The proliferation of books on the subject – such as those in the Addison Wesley Pattern Languages of Program Design series [Martin98]- is one indication of this.) The concept of software design patterns emerged from the ideas of patterns in building architecture, espoused by Christopher Alexander and documented in a number of his books, such as [Alexander79]. In these books, Alexander is looking at the successful and unsuccessful features of buildings – what makes people enjoy living or working in a certain space, why a particular arrangement of streets or looks appealing or fosters a sense of community. In doing this, he attempted to capture principles *explicitly* that may previously been *implicit* - in order that they can be reused. Figure 3 shows an example architectural pattern in the style of Alexander.

> *This pattern attempts to draw out the elements that make a courtyard successful ('living') or unsuccessful ('dead'). The successful courtyard on the left, has a flow of traffic through it, a focus of interest for someone sitting on the seat (e.g. a fountain) and a view and light provided through the exit at the top of the drawing. With the 'dead' courtyard, on the other hand, there is no real reason to enter the yard (no other exit), no focal point for people sitting on the sit and a lack of light through having three solid walls.*

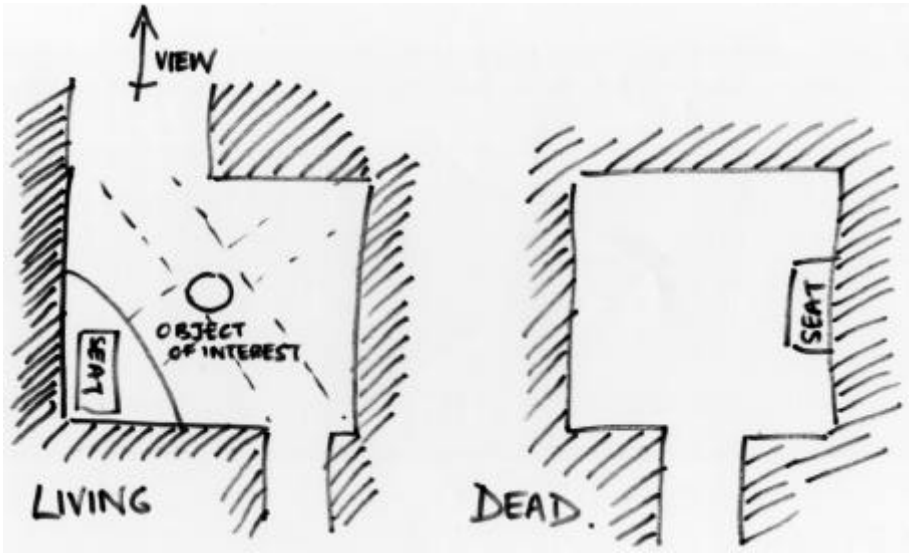


Figure 3 - A Building Pattern in the Style of Alexander

The design patterns community has attempted to relate such ideas to the construction of software – i.e. what makes a software architecture work well, be maintainable, be easily extended etc. Primarily this work has been in the Object-Oriented community but it is increasingly being recognised that the concepts are more widely applicable than just to OO designs. An important addition to the 'patterns' concept made by the Design Patterns community is the principle of capturing and recording the underlying rationale and principles of a pattern through structured documentation. Several formats have been proposed for this purpose – the format proposed by Gamma et al in [Gamma95] probably being the most widely adopted. This format suggests documenting design patterns under the following headings:

- **Pattern Name and Classification**
- **Intent**
- A**lso Known As**
- **Motivation**
- **Applicability (Necessary Context)**
- **Structure**
- **Participants**
- **Collaborations**
- **Consequences**
- **Implementation**
- **Example Applications**
- **Known Uses**
- **Related Patterns**

4. Safety Case Patterns

On examining the work of the Design Patterns community, we felt that the concepts could be readily applied to *safety arguments* to create *Safety Case Patterns*. These patterns, rather than addressing successful ways of putting buildings or software objects together, instead capture successful (i.e. convincing, well argued, easily understood etc.) argument approaches that are used within the safety case. As with Design Patterns, we wish to employ Safety Case Patterns as the medium for capturing:

- Solutions that evolved over time
- Company expertise
- Successful certification approaches
- 'Tricks of the trade'

The principal underlying Safety Case Patterns is to combine, the GSN described in Section 2 and the patterning concepts of the software design community. To do this required some extension of the GSN in

order that it could be used to support structural and entity abstraction (as, for example, OMT does for Design Patterns). Figure 4 shows a simple goal structure pattern that uses these extensions.

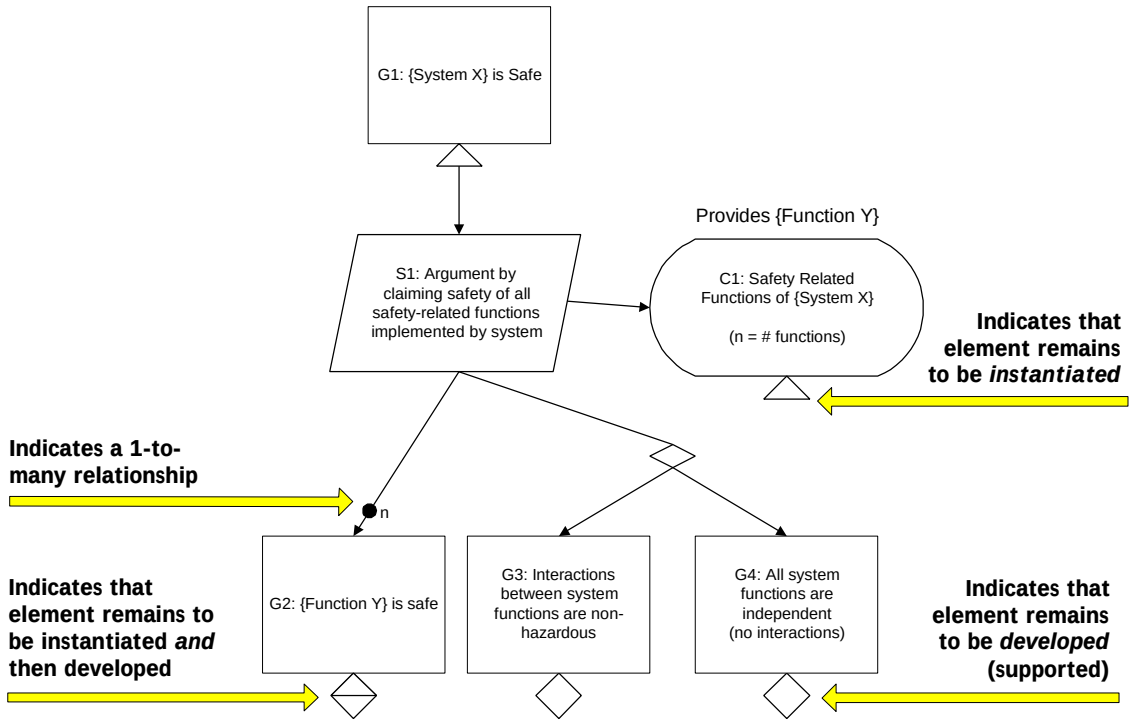


Figure 4 – Extensions to the GSN to Enable Pattern Description

This figure shows a goal structure pattern (without supporting documentation) representing a functional decomposition argument. In this structure, the top-level goal of system safety (G1) is re-expressed as a number of goals of functional safety (G2) as part of the strategy identified by S1. In order to support this strategy, it is necessary to have identified all system functions affecting overall safety (C1) e.g. through a Functional Hazard Analysis. In addition, it is also necessary to put forward (and develop) the claim that either all the identified functions are independent, and therefore have no interactions that could give rise to hazards (G4) or that any interactions that have been identified are non-hazardous (G3).

A Safety Case Pattern is not simply a GSN Pattern as shown in Figure 4. Additionally, there should always be a supporting pattern description (using the headings given by Gamma et al. in the previous section). To define patterns without clearly stating the underlying motivation and intent and without making clear where and (perhaps more importantly from a safety perspective) *where not* patterns should be applied could result in ignorant and inappropriate use of argument patterns within new projects. We provide a example of a fully documented safety case pattern as an appendix to this paper.

5. Our Experience To-Date with Safety Case Patterns

We have found that patterns can emerge at many different levels in a safety argument. We have found high level patterns such as the Functional Breakdown Pattern shown in Figure 4 and similar Hazard Directed Breakdown Patterns. These can be viewed as fundamental ('divide-and-conquer') approaches that exist within the armoury of approaches to constructing safety arguments. At the other extreme, we have also identified quite 'low level' patterns, e.g. capturing the types of claims that can be inferred from certain forms of evidence. We have also experienced that there are opportunities for both *horizontal reuse* (across domains) and *vertical reuse* (within a specific domain). An example of a domain-general pattern is a pattern for arguing adherence to Software Integrity Levels – the general concepts being applicable to a wide

range of industries. An example of a domain specific pattern is a pattern for arguing compliance with specific regulatory principles (e.g. aerospace requirement or nuclear safety assessment principle) that can't easily (or usefully) be transferred from one industry to another.

6. Summary

Common argument approaches exist between safety case developments. Informal reuse of safety case material already occurs – but in an uncontrolled, unpredictable and potentially dangerous manner. Based on the principles of Design Patterns combined with the safety argument representation concepts provided by the Goal Structuring Notation, we have developed the concept of *Safety Case Patterns*. Using these patterns it has been possible to identify and record reusable arguments in a number of situations and at different levels within the safety case. By creating a 'recipe book' of these approaches (our ongoing activity) we believe it is possible to capture expertise and improve safety case construction.

6. References

[Alexander79] The Timeless Way of Building
C Alexander
Oxford University Press, New York, 1979

[Gamma95] Design Patterns: Elements of Reusable Object-Oriented Software
E Gamma, R Helm, R Johnson and J Vlissides
Addison-Wesley, December 1995

[Martin98] Pattern Languages of Program Design 3 (Software Patterns Series)
Edited by R Martin, D Riehle, F Buschmann
1998, Addison-Wesley, ISBN 0-21-31011-2

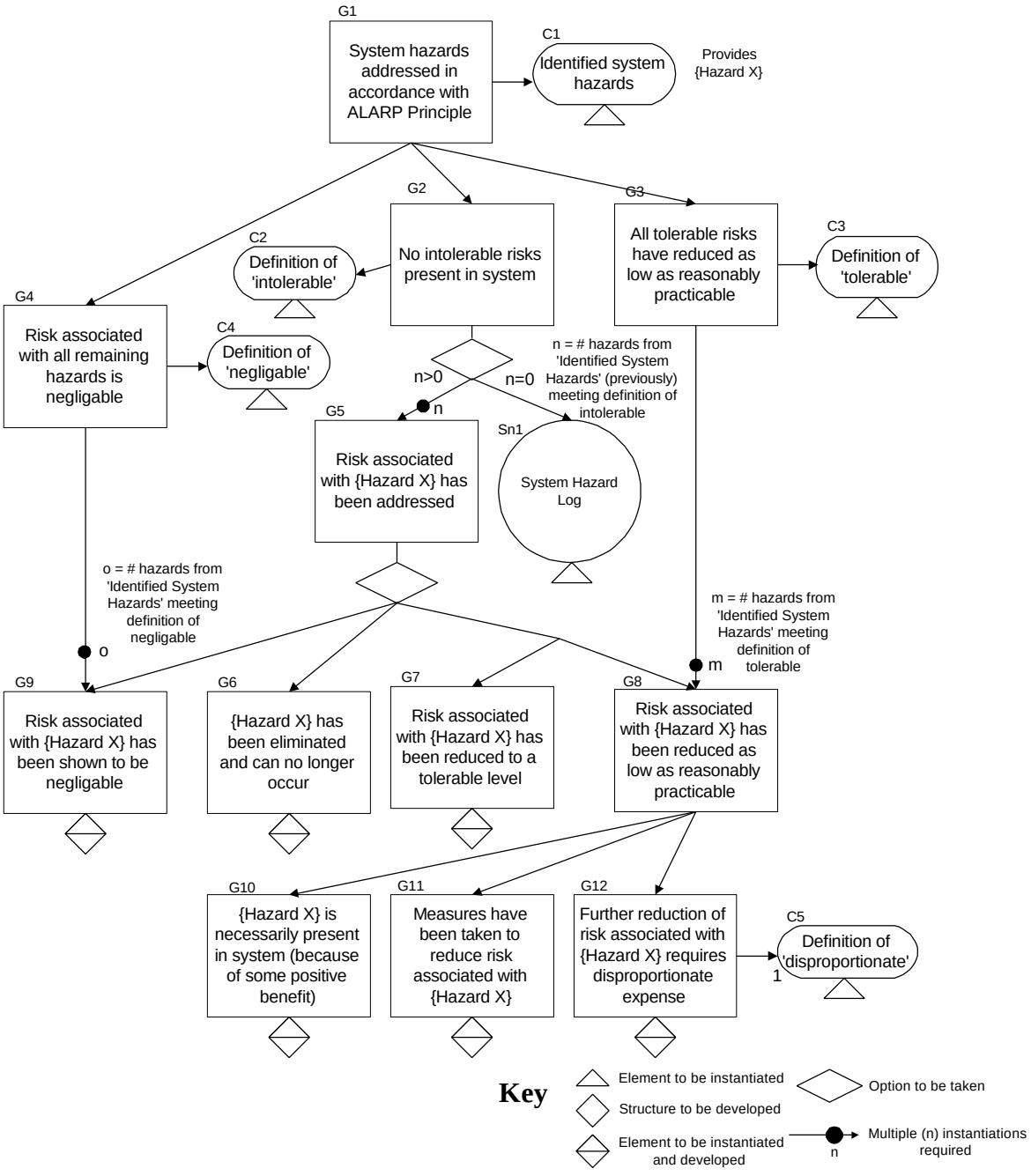
[Wilson95] Safety Case Development: Current Practice, Future Prospects
S. P. Wilson, T. P. Kelly and J. A. McDermid
in Proceedings of 12th Annual CSR Workshop, Bruges, Belgium 1995, Springer-Verlag

Appendix

ALARP (As-Low-As-Reasonably-Practicable) Safety Argument Pattern

Author	Tim Kelly	**Created**	04/02/97 10:41	**Last Modified**	05/02/97 09:47

Structure



<table>
<tr><td colspan="2">Intent</td><td>This pattern provides a framework for arguing that identified risks in a system have been sufficiently addressed in accordance with the ALARP principle.</td></tr>
<tr><td colspan="2">Also Known As</td><td>• Risk Reduction Argument Pattern</td></tr>
<tr><td colspan="2">Motivation</td><td>This pattern was developed for two reasons:
• To argue compliance with the ALARP principle at the highest level when addressing system level hazards.
• To provide a more structured approach to presenting a ‘Hazard Avoidance’ argument (See Hazard Avoidance Pattern) by showing differing treatment of hazards according to their associated risk.</td></tr>
<tr><td rowspan="9">Participants</td><td>G1</td><td>Defines the overall objective of the pattern</td></tr>
<tr><td>G2, G3, G4</td><td>Defines targets for three classes of identified risks: negligible, tolerable, and intolerable</td></tr>
<tr><td>Sn1</td><td>Provided at this point to support the claim that no intolerable risks have (ever) been identified with the system</td></tr>
<tr><td>G6 or
G7 and G8</td><td>Claims either that hazard has been eliminated or associated risk reduced to a tolerable level and dealt with as a tolerable risk.</td></tr>
<tr><td>G8</td><td>Defines ALARP target for each identified tolerable risk</td></tr>
<tr><td>G10, G11, G12</td><td>Claims required to support ALARP target:
• Hazard only acceptable if positive benefit achieved
• Risk reduction measures have been taken up to the point where further measures would be disproportionate to benefit gained.</td></tr>
<tr><td>G9</td><td>Claim for each remaining hazard that associated risk shown to be negligible</td></tr>
<tr><td>C1</td><td>A context identifying all system hazards, including indication of associated risks (e.g. Risk Category from A, B, C, D).</td></tr>
<tr><td>C2, C3, C4</td><td>A workable definition of ‘intolerable’/ ’tolerable’/ ’negligible’ risks that can be used as a basis for selection from the list of hazards (e.g. Intolerable = Risk Category A, Tolerable = Risk Category B or C, Negligible = D).</td></tr>
<tr><td>C5</td><td>The ALARP principle relies on some understanding of when it is no longer cost-effective to spend further money on risk reduction. This element, a definition of cost-effectiveness, is therefore required.</td></tr>
<tr><td>Collaborations</td><td colspan="2">An important aspect of this pattern is that it divides and conquers the goal of hazard mitigation / elimination according to the level of risk associated with each hazard. There are three strands to the safety argument: one tackling intolerable risks, one tackling tolerable risk and one discounting negligible risks. To satisfactorily support the top-level goal (G1) it is important that these three strands address all identified risks. The definitions of tolerable, intolerable and negligible (C3, C2 and C4 respectively) should therefore be so defined to cover and classify the range of possible levels of risks.
It should also be noted that the definitions of negligibility (C4) and disproportionate (C5) cannot be considered entirely independently. It would not make sense, for example, to force risk reduction to a level below that identified elsewhere as negligible.
As the goal structure shows, if the means of addressing a previously identified intolerable risk is to reduce it to a tolerable level, then the remaining risk must be tackled as for all tolerable risks. If the level of risk has been reduced to a negligible level, then the hazard must be tackled as a negligible risk.
It is important that the source of Identified System Hazards (C1) identifies the level of risk posed by a hazard in a way that permits sub-division into the classes of risk defined by C2, C3 and C4.</td></tr>
<tr><td>Applicability</td><td colspan="2">This pattern is applicable in contexts where the ALARP principle is accepted as the device for reasoning about the relative importance of risks and the cost-effectiveness of risk reduction.
In order to apply this pattern it is necessary to have access to the following contextual information:
• C1: Identified System Hazards
(See Participants section)
• C2, C3, C4: Definition of Intolerable / Tolerable / Negligible Risk
(See Participants section)
These definitions are typically provided by the appropriate regulatory authority, standards or through investigations by safety engineers, including discussions with customers.
• C5: Definition of Disproportionate
(See Participants section)</td></tr>
</table>

Consequences	After applying this pattern, there will be a number of undeveloped goals of the form: • **G7: Risk associated with {Hazard X} has been reduced to a tolerable level** • **G9: Risk associated with {Hazard X} has been shown to be negligible** • **G6: {Hazard X} has been eliminated and can no longer occur** • **G10: {Hazard X} is necessarily present in the system** • **G11: Measures have been taken to reduce risk associated with {Hazard X}** • **G12: Further reduction of risk associated with {Hazard X} requires disproportionate expense**
Implementation	Implementation of this pattern involves first instantiating the contexts C1, C2, C3, C4. In the context of the list of hazards referenced by C1, the solutions to goals G2, G3 and G4 can be provided. If no tolerable risks were ever present in the system, then reference to the system hazard log (Sn1) is sufficient to support the claim G2. However, if any intolerable risks have been identified, it is necessary to claim (G5) that these have been resolved through complete elimination of the hazard (G6), or reduction to a tolerable (G7, G8) or negligible (G9) level. For each tolerable risk identified an argument must be constructed (G6, G10, G11, G12) to demonstrate that it has been addressed in accordance with the ALARP principles. Measures taken in risk reduction must be stated in support of G11. Some evidence / argument of the non cost-effectiveness of further risk reduction measures must be supplied in support of G12, in accordance with the definition given by C5. Evidence of risk analysis (probably based upon consideration of probability of occurrence) is required in support of each claim of hazards posing negligible risk (G9). **Possible Pitfalls** • Not providing complete coverage of levels of risk through definitions C2, C3, C4 • Expressing definitions C2, C3, C4 in a way that is difficult to apply to the information provided by C1 (and vice versa) • Not having a commonly agreed concept of when to stop attempting further risk reduction (C1) - this can result in a non-uniform approach to tackling risks where significantly different levels of effort are committed to risks at the same level.
Examples	TBD
Known Uses	See *Industrial Press Safety Argument*
Related Patterns	Safe by Hazard Mitigation Argument