

Towards Formalising Reuse in Safety Cases

David Bush
National Air Traffic Services Ltd
T3G5, One Kemble Street
LONDON
WC2B 4AP

The contents of this paper reflect the views of the Author; they do not necessarily reflect the official view or policy of National Air Traffic Services Ltd.

Abstract This paper reports the results of an investigation into the scope for Reuse in Safety Cases in National Air Traffic Services Ltd (NATS). NATS undertakes major investments in safety related systems to meet its role of providing a safe and efficient air traffic service in UK airspace. The paper identifies three possible types of such reuse: Safety Requirements Reuse; Safety Case Structure Reuse; and Safety Case Artefact Reuse, and reports progress in achieving them. It identifies that Safety Requirements Reuse remains a hard research problem and that Safety Case structure reuse is currently practised in parts of the community; the paper therefore concentrates on the issues surrounding Safety Case Artefact Reuse.

The pre-requisites of such reuse are presented and the currently available tools and techniques are discussed against these pre-requisites. Further work is identified as necessary in the areas of safety case development tools; processes for safety case artefact reuse; the role of reviews and in customer/supplier relationships.

The paper illustrates how safety case artefact reuse can be made to fit into a typical systems engineering lifecycle, and how this in turn relates to the emerging safety case for the system. Examples are presented, for each stage of the lifecycle, of the activities of a Safety Engineer practising Safety Case Artefact Reuse.

INTRODUCTION

National Air Traffic Services Ltd. (NATS) plans, provides and operates a safe and integrated Air Traffic Management Service for the United Kingdom. Approximately 25% of NATS staff are engineers and about 70% of its fixed assets are in technical infrastructure. The recently completed Public Private Partnership (PPP) for NATS was driven by the future need for large-scale investment in technical infrastructure to match capacity with future demand. The successful recent opening of the New En-Route Centre at Swanwick is a welcome addition to NATS' capacity. However, in order to match expected demand, future investment must be effective and deliver technical systems that are, and can be shown to be, safe.

There is an increasing expectation and aspiration in NATS for future ATC systems to involve COTS components to a much greater extent, and possibly to seek strategic partnering arrangements with an ATC systems supplier. This arrangement would be such that a shared interest in the provisioning and ongoing development of a *family* of ATC systems is established for the long-term future.

Under such a strategy it would not be unusual for components (either individually or within systems families) to be used more than once across NATS. The potential for reuse of such components has prompted the question about the degree to which it is technically possible to reuse safety cases, or parts of them in order to achieve the most efficient procurement possible, and how such reuse might be incorporated into a Systems Development Process.

SAFETY CASES

Safety Cases Organisations responsible for developing safety critical and safety related systems have a legal and moral responsibility for satisfying themselves (and often their regulators) that the system to be put into operation is sufficiently safe. It is increasingly common for them to discharge this responsibility through the production of a Safety Case. A Safety Case, therefore, is intended to present a convincing argument that the system will be acceptably safe in operation.

TYPES OF REUSE WITH SAFETY CASES

General A general analysis of associated literature suggests that there are three forms of reuse related to safety cases:

- Reuse of safety requirements
- Reuse of the structure of safety case arguments
- Reuse of the artefacts of safety cases (claims, arguments, evidence)

Safety Requirements Reuse The primary responsibility for those designing safe systems is to remove all system hazards. This is not always possible, and so safety requirements are then derived for system hazards that cannot be so removed. They

define how the system should cope with remaining hazards. Many of these hazards depend strongly on the environment in which the system will operate, hence any attempt to reuse safety requirements must be robust against the environmental differences between implementations.

Safety requirements reuse is therefore the most fundamental form of reuse, and the scope for reuse of requirements seems at its most promising in the development of *product families*, where many requirements are shared by design. Nonetheless, even in this most promising circumstance there are challenges to straightforward reuse of safety requirements. (Lutz00) highlights that the analysis of requirements can differ between product line members (variations between family members may cause different interactions!) and (Clements98) highlighted as a research aspiration 'the ability to certify a set of safety-critical systems all at once'.

This suggests that even in the area where safety case reuse is most likely to work (i.e. product families), achieving it in practice is still considered to be an active problem even in the research domain.

Safety Case Structure Reuse (Kelly97) suggests that the reuse of patterns of safety cases is likely to be more rewarding than the reuse of artefacts of safety cases. With Goal Structured Notation (GSN) (Kelly00) presents a convincing case for the reuse of safety case structures both to provide absolute clarity of the strategy for demonstrating safety very early in the project, and a mechanism for allowing the development and refinement of arguments being developed as the project proceeds.

The GSN approach has been tried in NATS in a small number of projects. Although a full review of the utility of the approach is still to be completed anecdotal evidence suggest that it was successful in easing the communication of the overall structure of the safety arguments and for defining the strategy for the safety case.

Safety Case Artefact Reuse The remaining aspect of reuse in safety cases, and the main original thrust of this work was around the issue of safety case artefact reuse: reusing arguments claims and evidence.

(Kelly97) identifies the prevalence of informality in safety case artefact reuse and highlights potential problems both in the *process* (Safety Engineering) and its *product* (the Safety Case). The problems in the process arise because informal practise is ad-hoc and inconsistent in the organisation. Problems may arise in the product because the new context of the component may not be taken into account and no link is established between the original and reused safety

evidence - if the former were to be found in error, how would the latter find out about it?

PRE-REQUISITES OF SAFETY CASE ARTEFACT REUSE

To address the problems set by informal reuse, (Bush01) identifies a number of pre-requisites for formalising safety case artefact reuse:

- **Taxonomy:** Information elements that comprise a safety case, must be classified so that there is a common understanding of the information, and what that information means.
- **Information Model:** A model of the information, and rules for relating it must be defined. This must also allow the recording of the critical information in such reuse *context*.
- **Storage & Retrieval:** Safety cases must be stored, be retrievable, searchable and support tracability. Furthermore, reuse will be easiest if we can integrate the editing, storage and searching capabilities into a support tool.
- **Standardisation:** Safety cases must conform to the information model.
- **Information Availability:** Safety information must be available. For internal reuse, such a need may be confounded by legacy systems. In the COTS case, it is more difficult still to gain access to manufacturers' data.

PROGRESS AGAINST THE PRE-REQUISITES

Progress against these pre-requisites is achieved in recent significant work, this includes:

Safety Case Structure (Bishop98) provides a solution to the **Taxonomy** pre-requisite when it describes the key elements of a safety case as:

- **Claim** about a property of the system or some subsystem.
- **Evidence** which is used as the basis of the safety argument. This can be either facts, (e.g. based on established scientific principles and prior research), assumptions, or sub-claims, derived from a lower-level sub-argument.
- **Argument** linking the evidence to the claim, which can be deterministic, probabilistic or qualitative.
- **Inference** as the mechanism that provides the transformational rules for the argument.

Graphical Notations for Safety Cases Two common support tools for safety engineers involve the provision of graphical tools which allow the construction and editing of safety cases. These are Adelard's ASCAD and GSN (Kelly00). These notations are embodied in their respective tools, Adelard's ASCE and YSE's SAM2000. Both

approaches use graphical notations to distinguish between types of information, and constrain the relationships between the information types, they both provide a solution to the pre-requisite of having an **Information Model**.

Tool Support Although both the notations and tools described above have information models and tool support, the degree of support that such tools provide is currently a little too limited for them to fully support corporate reuse of Safety Case claims.

In particular the following functionality is still needed:

- **Network Support:** Tools will need to be capable of being used in a corporate network setting - with multiple users capable of accessing and editing information.
- **Traceability Support:** Tools will need to provide traceability from Safety Cases to the instances of reusable claims on which they call, audit trails of changes to claims and the connections made to them and comprehensive change notification mechanisms when referenced claims are changed.
- **Search and Retrieval:** Users will need to be able to search and retrieve claims stored in the tool in order to assess their relevance and veracity.

This is the sort of functionality readily available in tools supporting more mature areas of systems engineering, such as Requirements Management tools. Unfortunately Requirements Management tools do not have the sophisticated graphical front-end of the Safety case tools, although some vendors are considering such capabilities. Indeed, as Safety Case presentation, and the associated editing tools become more mature it is possible that vendors in both camps will seek to implement such capabilities if the demand exists.

Other Pre-Requisites Addressing the remaining pre-requisites falls much more clearly within the purview of the organisations who might wish to adopt such an approach. While **Standardisation** is helped by good tool support it also needs to be supported by reuse activities being included in the development process, and by the instigation of an adequate review and audit system to ensure the quality of compliance. The issue of **Information Availability** can be addressed internally by reverse engineering existing systems and their safety cases, although such an approach might fail a cost/benefit analysis. For external information sources the problem is potentially much harder. Vendor/Customer relations and the nature of the contracts agreed are the primary routes to availability in this context.

Progress against Pre-Requisites Progress against

these pre-requisites is shown in Figure 1, where solid lines represent an adequate solution and lighter lines show a need for further work.

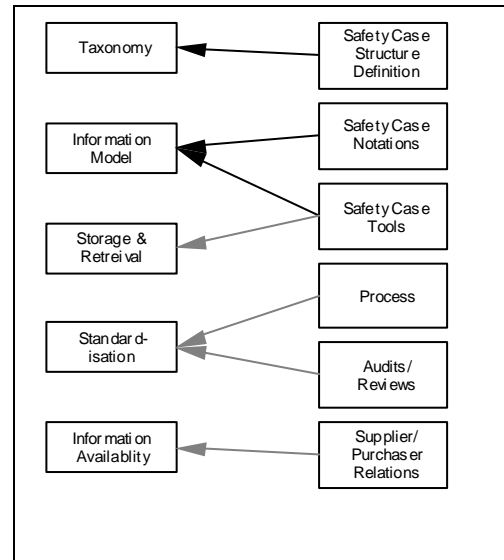


Figure 1 – Progress towards Safety Case Artefact Reuse

The remainder of this paper addresses how Safety Case Artefact Reuse could be fitted into a systems engineering process.

SAFETY IN THE SYSTEMS LIFECYCLE

Lifecycle Definition In order to present the reuse activities in context, we first define a lifecycle in which we can situate the activities. Given the preponderance of such models, and the passions they each engender, we have deliberately selected a very high level description of the activities needed. This should allow most people to map the activities we describe onto their own preferred detailed model. The model presented below is based strongly on that presented in (Leveson95), we define the main aspects of work in the lifecycle as: Concept Development; Design; Development; Production & Deployment; and Operation.

Safety Justification Similarly we can identify stages in the development of the claims, arguments and evidence for inclusion in the Safety Case. We summarise the breakdown of (Bishop01) and choose to label these as the Parts of the Safety Case:

- Part 1: Preliminary Safety Justification
- Part 2: Architectural Safety Justification
- Part 3: Implementation Safety Justification
- Part 4: Installation Safety Justification
- Part 5: Operation Safety Justification

The relationship between these two views is shown diagrammatically in Figure 2.

System Development Stage	Safety Plan/ Case Stage
Concept	Safety Plan & Strategy
	Part 1: Preliminary Safety Justification
Design	Part 2: Architectural Safety Justification
	Part 3: Implementation Safety Justification
Development	Part 4: Installation Safety Justification
Production & Deployment	Part 5: Operation Safety Justification
Operation	

Figure 2 - Safety in the System Lifecycle

Within this structure, we can now examine the typical actions that should be undertaken to implement a structured and defensible approach to reuse in safety cases. In order to provide a more concrete reference for this we first describe a possible scenario of reuse.

A POSSIBLE SCENARIO OF REUSE

Figure 3 shows the Safety Engineer for a new project developing the initial strategy for demonstrating the safety of the new system. This is shown as being carried out within a development environment. As well as providing the graphical tool for constructing the safety case, it provides the ability to browse the database of previously filed Safety Case artefacts (claims, evidence and argument for example), which may affect the way the Safety Engineer chooses to develop the strategy for this project.

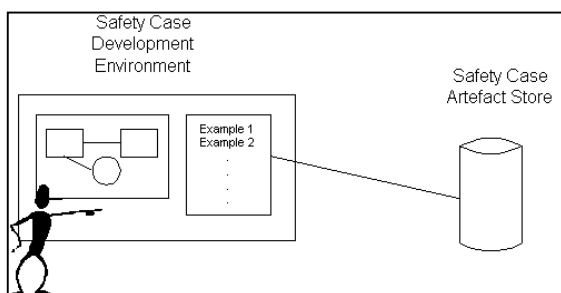


Figure 3 – Reuse Scenario I

The Safety Engineer decides that one particular artefact is likely to be of use in this project and so examines the detail in the database view window (Figure 4). As an example we shall assume that this reusable evidence is a claim about the reliability of the Microsoft Windows NT© implementation of a TCP/IP stack. The argument has two strands - the intrinsic reliability of the TCP/IP protocol (in red) and the reliability of the Windows NT© implementation (in blue). Also shown is the full

context information (assumptions, environment etc) about the claims, arguments and evidence (in green).

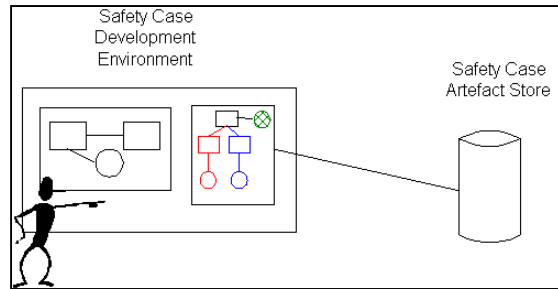


Figure 4 – Reuse Scenario II

From this the engineer identifies that, while this claim is useful to him, there are subtle differences in the context of use, and recognises that additional context information will need to be provided, possibly with additional supporting evidence to back them up. His new claim is therefore added to the project safety case – including the additional context information and evidence ((shown in dotted red). At the same time a link is made between the original claim in the database and the engineer's own safety case in which it is reused. This is shown in Figure 5.

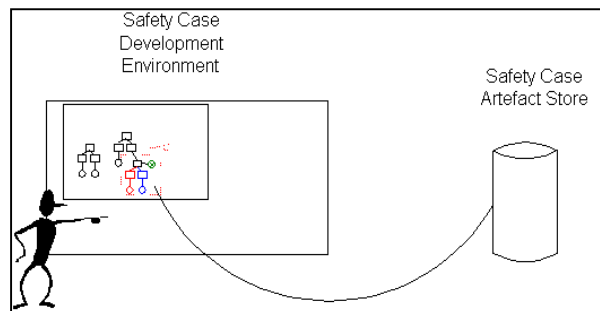


Figure 5 – Reuse Scenario III

It is possible at this stage that a number of systems (both in development and in service) now share artefacts stored in the database. We can imagine a scenario where an in service system begins to provide some long-term indication of the trustworthiness of the claim. We shall assume the worse case - that it is not as reliable as assumed.

As Figure 6 below shows, the traceability we have established from the database (rather than from paper based design documents) allows us to ensure that all systems are alerted to the concern. Each can then make a judgement as to whether this affects their project.

Further variations on such a scenario might show the developer adding claims to the database and so on. Work would be needed to establish the mechanism for this, and the trust which individuals might place in it. All these considerations would need to be identified and addressed were such a system examined.

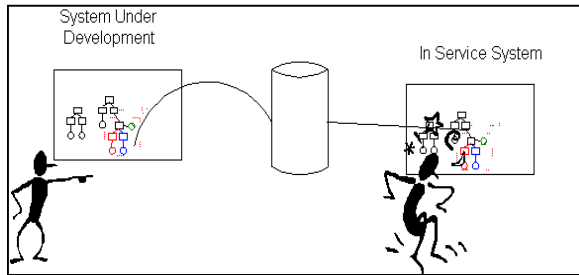


Figure 6 – Reuse Scenario IV

ARTEFACT REUSE IN THE SYSTEMS ENGINEERING LIFECYCLE

In this section, Figure 2 should be used as the selected frame of reference for safety activities in the systems lifecycle.

Concept The key activities in the concept stage are Safety planning, the derivation of safety requirements (targeted at the Part 1 Safety Case) and the development of an architectural solution through which responsibility for the satisfaction of the requirements can be allocated (targeted at the Part 2 Safety Case). The degree of formality in the endorsement of these parts of the safety case varies with circumstance; often safety regulators will require some oversight of them. In this stage, Safety Engineers can be expected to query the store of safety case artefacts (claims, arguments and evidence) in order to:

- Identify the likelihood of meeting certain safety requirements.
- Identify the degree of trust that might be placed in elements of the architecture.
- Identify the strength of evidence that is available for components.
- Identify successful argument strategies in similar cases.
- Deposit the safety case structure developed (and endorsed) as part of the safety planning activity.

The cases and examples provided by the repository will allow the creation of more robust strategies for demonstrating safety and the definition of systems architectures which are likely to be able to provide the integrity level required.

Design & Development The key activities at this stage are the creation and analysis of designs which will meet the allocated requirements, characteristic activities here might be the prototyping, design analysis, modelling and testing. This stage involves the development of the Part 3 Safety Case, although at this stage review of this is often less formal. At this stage, the Safety Engineer can be expected to use the repository in order to:

- Add evidence to support the existing arguments for the system under development.
- Draw upon previously deposited evidence and arguments for the emerging design
- Adapt existing information in the repository, update it for the specific project and add the updated information to the repository.
- Strengthen evidence in the repository as a result of design validation activities.
- Invalidate information in the repository as a result of design validation activities.

Production & Deployment The key activities in this stage include integration and testing activities, the deployment of the system and its preparation for entry into operational service. Where the deployment of the system could itself impact safety (for example by connection to an existing system for site acceptance testing or parallel running) this stage will involve a more formal review and endorsement of the Safety Case. At this stage, the Safety Engineer can expect to use the repository in order to:

- Add evidence to support the existing arguments for the system under development.
- Strengthen existing evidence in the repository as a result of testing and field service experience.
- Invalidate information in the repository as a result of testing and Field Service Experience.
- Update the ‘trustworthiness’ status of information in the repository where this information has been subject to a more formal endorsement and acceptance.

Operation This stage is characterised by activities which include the routine running of the system in its operation state; the collection of defect and fault information; forensic engineering activities as the result of major incidents; and the adjustment of the operational state and environment of the system. At this stage, the Safety Engineer can be expected to use the repository in order to:

- Update evidence information as the result of field service experience.
- Challenge information held on claims, arguments and evidence as a result of fault reporting or forensic engineering investigations.
- Respond to notifications provided by the repository, which might have been generated as the result of some other system altering information on which this system's safety case depends. Such responses might include:
 - Taking a system out of operation as a result of the notification of faulty evidence or argument.
 - The implementation of additional

architectural defences, such as manual checking, or procedural changes as a result of the notification of faulty information.

- The removal of some architectural defence as a result of the notification of some improved evidence or argument.

SUMMARY

This paper has reported the results of a NATS R&D study to investigate the scope for Reuse of Artefacts in Safety Cases. The paper identifies three aspects of reuse within safety cases; safety requirements reuse; safety case structure reuse (patterns) and safety case artefact reuse (claims, evidence etc.).

In the context of the pre-requisites for formalising safety case artefact reuse it examines the current state of tools and techniques and finds that while some pre-requisites can be met, further development of practice is required in the area of tools, process and customer/supplier relations.

Finally the paper combines a typical systems engineering lifecycle with a typical safety case development process and shows how the tasks and activities of a Safety Engineer practising safety case artefact reuse would fit in, and the use that engineer might make of the safety case repository at each stage.

REFERENCES

- (ASCE02) <http://www.adelard.co.uk/software/asce> accessed 4 Feb 2002.
- (Bishop98) Bishop, Peter and Bloomfield, Robin, "A Methodology for Safety Case Development" Safety-Critical Systems Symposium, Birmingham, UK, Feb 1998.
- (Bishop01) Bishop, PG., Bloomfield, RE., and Froome, PKD., "Justifying the use of software of uncertain pedigree (SOUP) in safety-related applications", HSE Contract Research Report 336/2001.
- (Bush01) Bush, D., "Reuse of Safety Case Claims - An Initial Investigation", London Communications Symposium, Sep 2001.
- (Bush02) Bush, D., "Reuse in Safety Cases", NATS R&D Report 0145, Feb 2002.
- (Clements98) Clements, PC and Weiderman, N., "Report on the Second International Workshop on Development and Evolution of Software Architectures for Product Families", *Carnegie Mellon University, Software Engineering Institute Special Report CMU/SEI-98-SR-003*, May 1998.
- (Kelly97) Kelly, TP. & McDermid, J., "Safety Case Construction and Reuse using Patterns", Proc 16th Int Conf on Computer Safety, Reliability and Security (Safecom '97)
- (Kelly00) Kelly, TP., "Arguing Safety - A Systematic Approach to Managing Safety Cases", PhD Report, University of York, 2000.
- (Leveson95) Leveson, N., "Safeware, System Saefity and Computers", Addison-Wesley, 1995.
- (Lutz00) Lutz, R., "Software Engineering for Safety: A Roadmap" in *The Future of Software Engineering*, ed Finkelstein, A., ACM Press, 2000.
- (SAM02) <http://www.yse-ltd.co.uk/> accessed 4 Feb 2002.

BIOGRAPHY

David Bush David works in NATS' R&D Group where he is responsible for carrying out research into Systems and Software Engineering activities. He is also an Engineering Doctorate Student at University College, London where his main research focus is on Requirements Engineering and Systems Architectures. He is a Member of the Institute of Management, a Chartered Engineer and Member of the IEE and is on the committee of the Requirements Engineering Specialist Group of the British Computer Society.