Outline Proposed Certification Approach

A. Simpson (Ebeni), S. Bull (Ebeni), T. Longhurst (CAAi)

This document explains how a logical argument approach can be applied to unify existing certification approaches from across the aviation industry and to provide a flexible means to achieve certification for new concepts and technologies.

Coordinator	L.J.P. Speijker (NLR)
Work Package Manager	B. Pauly (TR6)

Grant Agreement No.	314299
Document Identification	D1.3
Status	Approved
Version	1.2
Date of Issue	18-12-2013
Classification	Public

This page is intentionally left blank

Document Change Log

Version	Author(s)	Date	Affected Sections	Description of Change
1.0	A. Simpson et al.	22-11-2013		Version for approval by PMT
1.1	A. Simpson et al.	13-12-2013	ES, 1.3, 2.1, 2.3, 3.1, 3.1.1, 3.2, 3.2.3, 3.2.4, 3.4.1, 3.4.2, 5	Updated to address PMT comments
1.2	A. Simpson et al.	18-12-2013	ES, 1.2, 2.2, 3.1, 3.2, 3.3, 3.4.4, 4, 4.1, 4.2, 4.3, 4.5, 4.6(new), 5, C.4.3, C.4.4	Updated to address EASA comments

Review and Approval of the Document

Organisation Responsible for Review	Name of person reviewing the document	Date
NLR	L.J.P. Speijker, P.J. van der Geest, J.J. Scholte, A.L.C. Roelen, U. Dees	20-11-2013
CAAi	S. Long	20-11-2013
CertiFlyer	G. Temme, M. Heiligers	20-11-2013
Ebeni	J. Denness	20-11-2013
TR6	B. Pauly	20-11-2013
TUD	R. Curran, H. Udluft, P.C. Roling	20-11-2013
IoA	K. Piwek, A. Iwaniuk	20-11-2013
Avanssa	N. Aghdassi	29-11-2013
Isdefe	M. Martin Sanchez, I. Etxebarria	29-11-2013
APSYS	S. Bravo Munoz, J.P. Heckmann	05-12-2013
Organisation Responsible for Approval	Name of person approving the document	Date
TR6	B. Pauly	22-11-2013
NLR	L.J.P. Speijker	18-12-2013

Document Distribution

Organisation	Names
European Commission	M. Kyriakopoulos
NLR	L. Speijker, A. Rutten, M.A. Piers, U. Dees, P. van der Geest, A. Roelen, J.J Scholte, J.G. Verstraeten, A.D. Balk, E. van de Sluis
Thales Air Systems GmbH	G. Schichtel, J.-M. Kraus
Thales Air Systems SA	B. Pauly
EADS APSYS	S.B. Munoz, J.P. Heckmann, M. Feuvrier
Civil Aviation Authority UK	S. Long, A. Eaton, T. Longhurst
ISDEFE	M. Martin Sanchez, I. Etxebarria
CertiFlyer	G. Temme, M. Heiligers
Avanssa	N. Aghdassi
Ebeni	A. Simpson, J. Denness, S. Bull
Deep Blue	L. Save
JRC	W. Post, R. Menzel
JPM	J. P. Magny
TU Delft	R. Curran, H. Udluft, P.C. Roling
Institute of Aviation	K. Piwek, A. Iwaniuk
CAO	P. Michalak, R. Zielinski
EASA	K. Engelstad
FAA	J. Lapointe, T. Tessitore
SESAR JU	P. Mana
Eurocontrol	E. Perrin
CAA Netherlands	R. van de Boom
JARUS	R. van de Leijgraaf
SRC	J. Wilbrink, J. Nollet
ESASI	K. Conradi
Rockwell Collins	O. Bleeker, B. Bidenne
Dassault Aviation	B. Stoufflet, C. Champagne
ESA	T. Sgobba, M. Trujillo
EUROCAE	A. n'Diaye
TUV NORD Cert GmbH	H. Schorcht
FAST	R. den Hertog
SAE S-18	J. Dalton

Acronyms

Acronym	Definition
ACARE	Advisory Council for Aeronautics Research in Europe
ACAS	Airborne Collision Avoidance System
AIS	Aeronautical Information Service
ALARP	As Low As Reasonably Practicable
AMC	Acceptable Means of Compliance
ANS	Air Navigation Service
ANSP	Air Navigation Service Provider
AoC	Area of Change
ATM	Air Traffic Management
ATN	Aeronautical Telecommunications Network
CAA	Civil Aviation Authority
CCL	Common Certification Language
CNS	Communication, Navigation and Surveillance
COTS	Commercial Off The Shelf (System)
CS	Certification Specification
CSM	Continuous Safety Monitoring; Common Safety Method
EASA	European Aviation Safety Agency
EC	European Commission
EFB	Electronic Flight Bag
E-OCVM	European Operational Concept Validation Methodology
EU	European Union
FANS	Future Air Navigation System
FAST	Future Aviation Safety Team
FHA	Functional Hazard Assessment
FMS	Flight Management Systems
GASC	(Railway) Generic Application Safety Case
GPSC	(Railway) Generic Product Safety Case
GSN	Goal Structuring Notation
ICAO	International Civil Aviation Organization

Acronym	Definition
IM	(Railway) Infrastructure Manager
IMA	Integrated Modular Avionics
MET	Meteorological Data
OPENCOSS	Open Platform for Evolutionary Certification of Safety-Critical Systems
PSSA	Preliminary System Safety Assessment
RNP	Required Navigation Performance
RVSM	Reduced Vertical Separation Minima
SASC	(Railway) Specific Application Safety Case
SCDM	Safety Case Development Manual
SEooC	Safety Element out of Context
SESAR	Single European Sky ATM Research
SPI	Safety Performance Indicator
SRAC	Safety Related Application Condition
SSA	System Safety Assessment
STCA	Short Term Conflict Alert
TAS	Total Aviation System
UAV	Unmanned Aerial Vehicles
VNAV	Vertical NAVigation

This page is intentionally left blank

Executive Summary

This study is performed as part of the European Commission (EC) project ASCOS; this document presents an outline proposed certification approach for introduction of future changes to the aviation system. This document builds on previous work packages that reviewed current regulation and practice to identify *bottlenecks*[1] and *shortcomings* in the efficacy of the current processes, and examined current practice in the various *domains* of the *total aviation system* (*TAS*), identifying and evaluating possible options for improvement. In addition, further principles to be followed in development of the approach were identified.

A key conclusion of the previous work is that there is no single certification approach that can be applied universally within the *TAS*. Thus the proposed *certification* approach is to build on a framework using a logical *argument* for the *certification* of any change to the *TAS*, and supporting the overall top level *claim* that the change is acceptably safe. The *argument* is decomposed into supporting *claims* until the *claims* can be directly supported. The decomposition is aligned to the division of responsibility within the *TAS* and limited to that necessary to support definition of the interface between the *TAS* domains, and to dovetail with the existing certification approaches and specifications within each *domain*. Where existing standards are insufficient, this will support the definition of new specifications to support the introduction of novel technology or concepts. This framework advances the state of the art by driving unification of the *argument* across all *domains* and improving the rigour and consistency in the application of safety *arguments*.

Currently within aviation, *arguments* for safety (whatever form they might take) and the supporting evidence are distributed widely between various organisations, and often constructed in isolation. Re-integration of the safety *argument* is not always considered and essential information such as *dependencies*, *context*, *assumptions*, *constraints* or other assurance metadata can be lost. The proposed approach is to build an integrated argument for each proposed change to the system. As such an argument can become complex, the *argument* is structured into separate *modules*, to make the argument manageable. Each module encapsulates the *argument* for a particular component of the overall *argument*. The boundary of each *module* represents the public view of the *module* and includes a definition of the *claims* made in the *module* and associated *context*, *caveats* and *dependencies*. *Assurance contracts* are established between *modules* to capture the conditions which need to be satisfied in order to make an overall *argument*.

It is recognised that a given change may require endorsement from multiple authorities, each of which may only be competent to endorse the residual risk for part of the system. Thus it may not be possible for any one authority to endorse the top level of the *argument*. Consequently it is necessary, as part of the initial planning of the certification approach, to clearly define the parts of the *argument* which require endorsement by each authority. Effective application of the approach also requires an *argument architect* to take the overall responsibility for the development and maintenance of the *argument architecture* across all the affected

[1] Italicised text is used to denote terms defined in Appendix A.

domains, even if there is no one authority able to endorse this overall argument. Further work is needed to identify who is best placed to undertake this *argument architect* role.

In the proposed approach each *module* is developed in the context of the whole safety *argument* and makes *claims* and establishes essential boundary dependencies and caveats that are visible without the need to read detailed safety assessment documentation. This allows the interdependencies between parts of the system (e.g. different *domains*) to be clearly defined and managed.

The framework can readily incorporate existing certification approaches and evidence hierarchies, and be used to establish the necessary interactions between individual *domains* and organisations. This allows the maximum reuse of existing certification approaches (e.g. the application of current standards such as ARP4754A/ED79A) where these remain applicable and enables the integration of different approaches taken in different *domains* by ensuring the dependencies between each are clearly defined and managed.

The proposed approach is not dependent on a specific representation or tool: module safety *arguments* can be developed using text or a variety of graphical notations allowing for example, easy adoption of current approaches.

The proposed approach is sufficiently flexible to incorporate all the options recommended by the previous work package: it allows retention of existing *certification* processes within individual *domains*, while also ensuring that the *context* in which the existing certification is developed is fully considered within the overall *argument*. The flexibility also allows for alternative approaches to be taken where the change being introduced is not covered by existing specifications, thus supporting innovation in process or technology, as required by the overall aims of the ASCOS project. This may involve changes between *performance-based* and *compliance-based* approaches; it may also introduce approaches from other aviation domains or from other industries. It also provides the flexibility to introduce the *proof of concept* approach.

The proposed approach has taken into account industry concerns including the identified *bottlenecks* and *shortcomings* and the directly expressed concerns of the ASCOS User Group. Although it is not possible for the approach to directly resolve all the issues identified, it does provide a framework which supports them being addressed. In particular, the approach:

- provides flexibility to support innovation in (a) technologies and concepts and (b) certification approaches;
- provides a framework to improve communication and integration between domains;
- defines step-by-step process (see section 4) which encompasses the whole lifecycle and supports engagement of stakeholders (including authorities) throughout;
- builds on the approach adopted by EUROCONTROL and further developed by the SESAR research programme.

The approach is supported by generic *argument* templates from which detailed *arguments* can be developed. The initial template proposed for application in the ASCOS case studies is a high level *argument* successfully

used in ATM applications (see section 3.2) and which provides a flexible framework for integrating safety arguments across domains. This template *argument* is structured to address the whole development lifecycle, from development of a specification, through to monitoring of the system throughout its operation. Additional templates have been developed, and these will be refined during the case studies.

The logical *argument* approach is well established in the ATM *domain* and in other industries and shows the most promise for achieving the aims set by the previous work. However the approach is not intended to replace other well established processes within the *TAS*, but rather to use the argument framework to integrate the results of the approaches taken in each domain.

However, as there is no common or widely agreed methodology for constructing such arguments and there is no clear practice on interfacing arguments between domains or between lifecycle phases, the application proposed here seeks to introduce a number of innovations which will drive forward the state of the art, to:

- provide a basis for unification of multiple *certification* approaches from multiple *domains* within a single logical *argument*;
- encourage wider application of logical *arguments* to address novel systems and concepts,
- develop a more robust approach to safety argument construction;
- utilise safety arguments to support the cross fertilisation of certification approaches;
- apply the principles behind safety argument *modularisation* to manage interfaces between different parts of the system and *lifecycle phases*.

The steps for application of the approach identify the activities to be undertaken throughout the lifecycle of the change, from the initial definition, through the planning and execution of the certification approach, including engagement with the relevant authorities. Triggers for application of the approach are defined.

The steps also consider the transition when the proposed change is put into operation and the continuous monitoring of the changed system to ensure that the claimed level of safety is indeed met. The guidance includes the interaction with the ASCOS work packages:

- WP3 (safety risk management) which provides the safety assessment methodology framework to support the (*a priori*) risk assessments required to support the overall argument;
- WP2 (continuous safety monitoring) which provides the monitoring framework to support the (*a posteriori*) risk assessment of the change in operation.

The proposed approach will be applied to the ASCOS case studies (WP4) and validated (WP5) through comparison with the approach taken in previous certifications within the aviation industry. This experience will subsequently be used to refine the proposed certification approach and develop further guidance material to allow application of the approach across the *total aviation system*.

This page is intentionally left blank

Table of Contents

List of Figures

List of Tables

This page is intentionally left blank

1 Introduction

1.1 Background

Fundamental changes in the institutional arrangements for aviation regulation in Europe, the introduction of new technologies and operations, and demands for higher levels of safety performance, call for the adaptation of existing certification processes. The European Commission (EC) Project 'Aviation Safety and Certification of new Operations and Systems' (ASCOS) contributes to the removal of certification obstacles and supports implementation of technologies to reach the EU ACARE Vision 2020 [2] and Flight Path 2050 [45] goals.

The main objective of the ASCOS project is to develop novel certification process adaptations and supporting safety driven design methods and tools to ease the certification of safety enhancement systems and operations, thereby increasing safety. The project will follow a total system approach (see Appendix B), dealing with all aviation system elements (including the human element) in an integrated way over the complete life-cycle. ASCOS is also tasked with ensuring that any proposed approach is cost-effective and efficient.

Previous ASCOS work provided an overview on current regulations and the degree to which these regulations are implemented within the aviation community [1]. It also examined accident statistics and trends within the European aviation domain. Potential *bottlenecks* and *shortcomings* in the efficacy of current regulatory or *certification* processes were identified. Whilst not exhaustive the assessment was also supplemented by a review of other material highlighting *bottlenecks* and *shortcomings*. Previous ASCOS work also examined the current practice in the various *domains* of the *total aviation system* (*TAS)* and identified eight possible options for improvement of the certification process; the benefits of each option were then evaluated against a number of criteria [1]. The options which were considered to have the most promise are:

- Option 2: Change between performance-based and compliance-based or vice versa
- Option 6: Proof of concept approach
- Option 7: Enforce existing rules and improve existing processes
- Option 8: Cross-domain fertilisation

Instead of concluding on a single best option for certification process adaptation, several principles to be followed in the development of the proposed certification approach were identified:

- Avoid unnecessary change, recognising the good approaches already in place.
- Provide a generic *certification* framework encompassing the *Total Aviation System* (*TAS*).
- Use a common language across all *domains* based on safety *argument* concepts (e.g. argument-based as used in OPENCOSS), allowing flexibility to accommodate a variety of approaches across domains.
- Provide rigorous management of interfaces, both between *domains* and between the *TAS* and its *external environment*, to ensure that all key safety issues are properly addressed and not lost at interfaces.
- Allow, within each *domain*, the proposed *certification* approach to evolve from the current approach by

  - keeping the existing approach where no change is required
  - learning lessons from other *domains* where this gives improvement
  - ensuring that *bottlenecks* and *shortcomings* are addressed by the proposed approach.
- Promote flexibility within each *domain* to allow introduction of new technologies or procedures.
- Harmonise approaches between *domains* where this is advantageous or necessary.
- Simplify *certification* processes, where there are:
  - demonstrable benefits and
  - no loss of confidence in the assurance of safety.
- Reinforce existing techniques where they are appropriate but not consistently applied.
- Provide a mechanism for identification and resolution of further *bottlenecks* and *shortcomings*.
- Introduce a bridge between the regulations in different *domains* where needed.
- Take into account the electronic hardware more explicitly in the proposed approach.
- Consider the fact that less experience is gained by the flight crew when more automation is used.
- Promote adoption of proposed approach by international authorities.

1.2 Objectives

This study covers the initial outline development of the proposed certification approach.

Therefore, the main aim of this study is to develop a proposed certification approach for the *total aviation system* (*TAS*), offering improvement over the existing certification/approval processes used within the *TAS* in terms of[2]:

- Efficiency in terms of cost and time.
- Ability to analyse and demonstrate acceptable safety for new concepts and technologies.
- Ability to analyse and consider the entire aviation system rather than sub-elements in isolation.

Within the last bullet above, it is recognised that interdependence between the sub-elements of the system (*domains*) plays a significant part.

The outline proposed certification process provides sufficient detail for the proposed approach to be applied during the case studies in WP4, and for stakeholders to evaluate the process and to demonstrate that the process[3]:

- Is practical.
- Will be able to cover the wide range of proposed changes within the total aviation system.
- Can provide an adequate level of safety assurance.

[2] As defined in the ASCOS DOW [35].
[3] As defined in the ASCOS DOW [35].

1.3 Approach

The proposed approach is to develop an initial outline proposed certification approach based on the principles identified as a result of previous work [1], and as summarized in section 1.1.

This initial proposed approach will then be applied and evaluated during a number of case studies within ASCOS WP4 and validated as part of ASCOS WP5. The results of these activities will then be used, together with input from the ASCOS User Group, to produce a further refined approach capable of application across the *total aviation system*.

From the previous findings [1], it is apparent that:

- Different *domains* of the *TAS* have very different approaches to *certification* and that, in the main, these approaches work well in their respective *domains;* but there is no panacea approach. It is clear that any adaptation of the *certification* approach must not lose the benefit or assurance provided by the existing approaches; instead the best parts of these approaches must be retained, while also providing the enhancements needed to ensure efficient cross-domain *certification* of new concepts and technologies.
- When new technologies or concepts are introduced, current standards are unlikely to be sufficient, especially when *certification* takes a *compliance-based* approach. Thus an approach is needed to support efficient *certification* in the absence of existing technology-specific standards; although the approach may lead to the development of standards for future deployment of similar technology.
- Interfaces present a particular concern because, where issues[4] are transferred between systems or organisations, it is easy for them to be lost, overlooked or forgotten: where such issues are safety-related, any such failure has the potential to affect the safety of the overall system. Thus, interface identification and management is a key area of focus for the revised certification approach.

Guided by these concerns, this study seeks to retain as much as possible from existing practices within a framework that manages interfaces and facilitates introduction of new technology and the transfer of good *certification* practices. In this way the approach takes into account the most promising options for certification process adaptation – further detail on this is given in section 3.4. Additional consideration is given to how the approach could be used to address other key *bottlenecks* and *shortcomings* (as described in section 3.4.3). However, it is not within the scope of this study to address every concern raised.

The approach taken is to propose a logical safety *argument* framework which is used to demonstrate that the proposed changes to the *TAS* are acceptable. Over time this framework grows to encompass the *TAS*. The framework sits outside the approach taken by any individual *domain* and provides sufficient flexibility to unify the approaches taken in each *domain* without imposing unnecessary changes to the existing *certification* practices and to provide the link between the standards and the overall *argument* being made. The approach

[4] In this context issues include *assumptions*, *limitations*, *dependencies* or other *constraints* and caveats that explicitly or implicitly cross the interface between one party and another in the *TAS*.

also provides a framework through which the issues at the interfaces between *domains* can be rigorously managed. This approach has been successfully applied in industry (see sections 3.1 and 3.2). It is recognised that the approach is not currently used in all domains; however the approach provides the flexibility both to encompass existing certification approaches where appropriate and to provide an alternative where the existing approaches do not provide the necessary tools. The ASCOS case studies provide the opportunity to demonstrate and refine the application of the approach in unifying certification arguments across the *TAS*.

The framework is used to capture the *context* associated with a change to the aviation system and to determine the most appropriate approach to be taken for the specific change. Where appropriate, the existing approach within the relevant *domain(s)* can be followed, but the framework allows an alternative approach to be chosen where this is necessitated by the change (for example, to allow the case to be made for a novel technology where this is not covered by existing Certification Specifications).

The use of safety arguments is already established within parts of the *TAS* and elsewhere. It is argued herein that this approach shows the most promise for achieving the key principles set out by previous work [1]. However, currently there is no common or widely agreed methodology for constructing such arguments, and there is no clear practice on interfacing of aviation safety arguments between domains, sub-domains or between lifecycle phases e.g. design and operations. There are also documented examples of poor application [11] or use of fallacious safety arguments [36]. The proposed approach is thus seeking to:

- provide a basis for unification of multiple *certification* approaches from multiple *domains* within a single logical *argument*;
- encourage wider application of logical *arguments* to address novel systems and concepts,
- develop a more robust approach to safety argument construction;
- utilise safety arguments to support the cross fertilisation of certification approaches;
- apply the principles behind safety argument *modularisation* to manage interfaces between different parts of the system and *lifecycle phases*.

The *argument* for a complex change is likely to span multiple *domains* and thus become large and complex. The framework allows the argument to be broken down into *modules*, aligned for example with current boundaries of responsibilities and system architectures, and each of which can be developed separately. Each module may take very different approaches, mirroring the approaches taken in each domain by the different organisations involved.

The decomposition into *modules* includes the identification and management of interfaces, thus ensuring that relationships between *modules* are fully considered throughout the development of the argument and are not lost. The approach considers the whole lifecycle, thus helping to prevent the loss of interface issues when for example implementation of a change is complete.

Modularisation also allows the approach in a particular part of the argument to be changed without affecting the rest of the argument, as long as the affected module still complies with the interface(s) established with the rest of the argument.

The aim of this work package is to develop the approach to a level which allows its application to the ASCOS case studies (i.e. WP4). These case studies will be used to refine the approach so that it is suitable for wider application.

Note that it is also recognised that the approach exists in the context of what could be viewed broadly as a three tiered regulatory[5] framework:

1. Regulation and legislation
2. Competent authority approach to compliance / acceptable means of compliance
3. Industry standards, recommended working practices, guidance

This framework exists in all domains of the *TAS* although there are minor differences and the boundaries between the tiers can be blurred.

In this context the primary focus of the proposed approach is to consider how to improve the competent authority approach to compliance; however, WP1 may make recommendations for changes to regulation or standards where this gives a significant benefit through harmonisation or simplification.

1.4 Structure of this Document

This section (section 1) presents the background and the rationale for the approach taken in development of the proposed certification approach.

Section 2 presents the concept of logical *argument* and explains how complex *arguments* can be decomposed into *modules*. This section also introduces a notation (GSN) supporting presentation of, and reasoning about, logical *arguments*.

Section 3 shows how the logical *argument* concept introduced in section 2 can be applied to the *Total Aviation System* (*TAS*), using the example of a generic *argument* widely used within the ATM *domain*. As explained, this approach can also be readily adapted to other *domains*. This section discusses how this approach can be broken down into logical *modules* with well-defined interfaces to reflect the different *domains* of the *TAS*. This section also shows how the options from previous ASCOS work are addressed within this logical *argument* framework and explains how the *bottlenecks* and *shortcomings* identified in previous ASCOS work are addressed.

Section 4 gives a step by step guide to application of the approach to the case studies. It should be noted that, at this stage, the approach has been developed sufficiently to permit application to the case studies. Experience from the case studies will be used to refine the approach sufficiently to permit wider application.

Section 5 presents the conclusions of this study.

[5] Including international regulation and legislation

The main body of the document (as described above) is supplemented by multiple appendices which provide detailed evidence and supporting information as follows:

- Appendix A: Glossary of terms – different *domains* within the aviation industry use different terminology – an argument covering the *TAS* must use a consistent set of terms throughout. Terms defined within this appendix are shown in italic text throughout the document.
- Appendix B: Total System Approach – presents an outline explanation of the concept of the *total aviation system* as well as a model of the *TAS* in order to provide context for the discussion of interfaces between *domains*.
- Appendix C: Questionnaire Summary – as part of the research for this study, a questionnaire was distributed to partners of the ASCOS programme and to other interested parties, including members of the User Group; in addition, interviews were conducted with various stakeholders. This appendix presents a summary both of the rationale for the questionnaire and of the results.
- Appendix D: Template Arguments – presents outline *argument* structures for the most commonly used *certification* approaches, including both *performance based* and *compliance based* approaches as well as *proof of concept*.
- Appendix E: Related Approaches Across Industry – presents relevant material on how *certification* is approached in other industry sectors.
- Appendix F: Fallacious arguments reading list – section 2 explains that logical arguments need to be carefully constructed and reviewed to ensure that they are correct. This appendix gives references to research and analysis into how the approach can go wrong.
- Appendix G: Key to Goal Structuring Notation – GSN is a notation used to express logical *arguments*; this appendix provides a key to ensure that these can be read and understood.

1.5 Typographic Conventions

Italic text is used for terms with specific meanings defined in Appendix A.

> Indented paragraphs in this font are used for examples drawn from the questionnaire used to gain input from a wider group of stakeholders as described in Appendix C.

2 Applying Logical Argument

2.1 Introduction to Logical Argument

All *certification* approaches or routes to acceptance are based on an *argument* of some form. This may be an *implicit argument* effectively defined by the procedures to be followed to gain approval, or it may be an *explicit argument* presented in the approval submissions, e.g. by constructing a safety case. In some domains the argument can consist of explicit and implicit components, for example the explicit requirements in a Certification Specification and the often implicit assumptions or context used in deriving those requirements.

An argument consists of:

- A set of *claims* that express why a system or service (made up of equipment, people and procedures) is considered to be acceptable.
- Supporting information (*strategy*, *context*, *assumptions* and *justifications*) which explains the reasoning behind the *argument*.
- Supporting *evidence* to substantiate the claims at the lowest level in the argument (i.e. those which are not further decomposed within the *argument*).
- Caveats or *conditions* that constrain or limit the interpretation and further application of the claims made.
- Dependencies on other components of the aviation system outside the bounds of the system or service change under consideration.

An *argument* is presented as a hierarchy below a top level *claim*, usually of the form "System X is acceptably[6] safe." The top level *claim* is decomposed into a hierarchy of *sub-claims*: at each level of the *argument*, satisfaction of a *claim* is demonstrated by the satisfaction of all the *sub-claims* into which it is decomposed.

An advantage of *explicit arguments* is that they can help to avoid some of the pitfalls faced by *implicit arguments*. One particular example is where a specification is based on *assumptions* about the *context* in which equipment will be used or about the technology which will be used to deliver to the specification. If these *assumptions* are invalid, equipment which meets the specification may still prove to be unsafe, because the overall *argument* is fallacious. An example is the changing role of an aircraft's FMS with the introduction of advanced functions such as advanced RNP: the *argument* for introduction of such a function needs to consider whether the existing FMS specification is sufficient to safely support the new function or whether adaptations are required which fall outside the existing specification.

Safety cases in the ATM *domain* often take the logical *argument* approach to present the argument. In fact, such safety cases are often presented in parts with the parts corresponding to the first tier of sub-*claims* in the generic *argument* presented in section 3.2.

[6] In this context acceptability is taken to imply that evidence satisfies relevant regulation and legislation such that the appropriate safety authority can approve, licence or otherwise certify the system or service.

It is important to be aware of the possible shortcomings of logical *argument*. Mistakes can be made or poor reasoning can be used in the construction of *arguments*, resulting in fallacious *arguments*. *Arguments* can be fallacious whether or not the conclusions are true. Fallacious arguments are discussed further in section 2.4.

The logical *argument* approach is not without its critics. A recent example is the Haddon-Cave investigation into the loss of a Nimrod aircraft over Afghanistan [11] which levelled a number of criticisms at the use of safety cases. This is not a criticism of the use of *argument* per se, but is a criticism of the way in which this approach has been poorly applied. In particular, Haddon-Cave suggested that safety cases[7] should be:

- succinct
- home-grown
- accessible
- proportionate
- easy to understand
- document-lite.

A rigorous approach to *arguments* helps to achieve these objectives. Some ways in which this is achieved is through ensuring that the argument remains focused on the goal and by ensuring that the *argument* is structured using precise definitions so that it is easy to follow and to reason about. Arguments must not be made more complex than necessary[8], and should adopt existing approaches (such as demonstration of compliance with existing standards) at the highest level possible.

However, there is also a need for rigorous review to ensure that the *arguments* are correct.

Logical *arguments* generally take one of three forms:

1. process based (the applicant demonstrates that they have followed a particular process);
2. product based (the applicant demonstrates that the product meets a specification);
3. objective driven (the applicant demonstrates that particular objectives or performance criteria are met – e.g. safety targets).

The *argument* must be supported by appropriate *evidence*. Supporting *evidence* is often categorised as *direct evidence* or *backing evidence* (as described in the EUROCONTROL Safety Case Development Manual (SCDM) [15]):

- *direct evidence* – *evidence* that a particular objective has been achieved (i.e. that a higher level *argument* or *claim* has been satisfied). This is evidence relating directly to observable propertyes of an output or product (i.e. the output of a process).
- *backing evidence* – *evidence* that there is sufficient confidence that the *direct evidence* can be relied upon (or is "trustworthy"). This is *evidence* relating to the properties of the processes by which *direct*

[7] Haddon-Cave also recommended that the documents should be renamed "Risk Cases".
[8] However, complexity of the argument is often driven the by complexity of the system (in the widest sense of the term) about which the argument is being made, as further addressed in section 2.2.1.

evidence was obtained, e.g. tools and techniques, human resources applied were qualified/competent and properly deployed.

2.2 Modularisation of Arguments

Arguments can become complex, especially when the systems about which they are made are complex or large as in the case of the *TAS*.

In order to make such *arguments* manageable, they need to be split into smaller sub-*arguments*. The key principle is to split the *argument* into well-defined *modules*, with well-defined interfaces so that these *modules* can be developed separately from one another in confidence that the final result will be a consistent and correct overall *argument*. This approach is analogous to similar principles in software and system design.

The modular approach to safety *arguments* was developed to support the concept of Integrated Modular Avionics (IMA), which uses an integrated architecture with application software portable across an assembly of common hardware modules. The concept has been applied both in military and civil aircraft, including the Airbus A380, Boeing 787 and F-22 Raptor. The modular approach has also been applied in the automotive industry. The approach has also been researched within the OPENCOSS programme, see Appendix E.1.2.

The following papers have been published on the modular approach:

- Concepts and Principles of Compositional Safety Case Construction [16]
- A Case Study on Safety Cases in the Automotive Domain: Modules, Patterns and Models [19]
- Safety case architectures to complement a contract-based approach to designing safe systems [20]
- Safety Case Composition Using Contracts – Refinements based on Feedback from an Industrial Case Study [21]

The principles of modularisation are useful in the ASCOS environment because they allow:

- Development of a *certification* approach / *argument* covering the *total aviation system* with clear definition of the interaction between different elements of the system.
- Compartmentalisation of different parts of the *argument*, allowing updates to parts of the *argument* without affecting the rest, as long as the *argument* is unchanged at the interface of the *module*.
- Reuse of parts of the *argument* without requiring extensive redevelopment, again as long as the interface with other parts of the *argument* remains unchanged.
- Individual parts of the *argument* to adopt the practices habitually used in the domain while also ensuring that the argument can be integrated with the rest of the *total aviation system*.

Modularisation of arguments is already explicitly supported in some industries. The rail industry (EN50129 [7]) uses the concept of generic safety cases, which document the *argument* and *evidence* that a particular product or system is safe in the context of a number of assumptions about the *external environment* and the use of the product and conditions (Safety Related Application Conditions - SRACs) on its application. The safety *argument*

is then valid for use of the product in any application, as long as the assumptions are (demonstrably) valid and the conditions are met. (See Appendix E.2.3.)

The automotive industry uses a similar concept of Safety Element out of Context (SEooC), where a component is developed for some foreseeable hypothetical application. This new component can be re-used in a variety of (different) contexts, subject to provision of the required justification and validation as well the appropriate revision of the safety plan accordingly. When developing or reusing a SEooC, some of the safety lifecycle activities are tailored (ISO 26262-2 [8], Clause 6.4.5.6) to avoid unnecessary replication of the activities.

The notion of cross-acceptance (see Appendix E.2.3) is where equipment already accepted and in service under a particular authority (e.g. the competent authority of a particular state) is accepted for use under another authority (e.g. in a different state) without the need for reassessment to support the new certification. (In practice this only works for the generic product, and the new authority will still need to establish that the application in the new environment meets any specific requirements.)

Modular certification is already available for simple airborne components under the European Technical Standard Orders (ETSO) scheme. These authorisations are issued under Part 21, Section A, Subpart O of EC/748/2012 [57]. This certification provides a step towards use of these components, although it is then necessary to additionally apply for approval on board specific aircraft types. Certification has been granted to around 200 components under this scheme. More details of this scheme can be found on the EASA website [58]. There is currently a rulemaking task to develop this approach for Integrated Modular Avionics.

> Accidents resulting from air frame icing (of Fokker aircraft) while on the ground led to an overhaul of Fokker's operational rules with the target of ensuring that aircraft are clean before take off. The manufacturer had to design a solution which met the differing requirements of three different authorities (in USA, Canada and Netherlands). The solution was driven by the manufacturer (perhaps because it is their name which will be discredited wherever any accident occurs), although the responsibility for implementation depends on interaction between airline, manufacturer and aerodrome procedures. This example illustrates how issues can cross boundaries both between domains and organisations, and how important it is to ensure that communication across these boundaries is effective. The modular argument approach described in this document captures these cross boundary issues and supports effective management of them. A logical argument approach (supported by all stakeholders), if taken from the outset, could also have identified a different ideal solution, by changing the responsibility for the ground de-icing task.

2.2.1 Modular Safety Argument Architecture

Modularisation facilitates the breakup of complex or large *arguments* into manageable *modules*. Each *module* encapsulates the *argument* for a particular component of the overall argument. The boundary of the *module*

represents the public view[9] of the *module* and includes a definition of the *claims* made in the *module* and associated *context*, caveats and *dependencies*. The *module* boundary definition provides all the information necessary to facilitate linking with other *modules*.

As the example Figure 1 below for a ground vehicle shows, even a modular safety *argument architecture* can still be very complex but only because the system it is addressing is complex. Modularisation provides a sound basis for identifying the inter-module links that do or should exist, and making sure these links are valid and functioning.

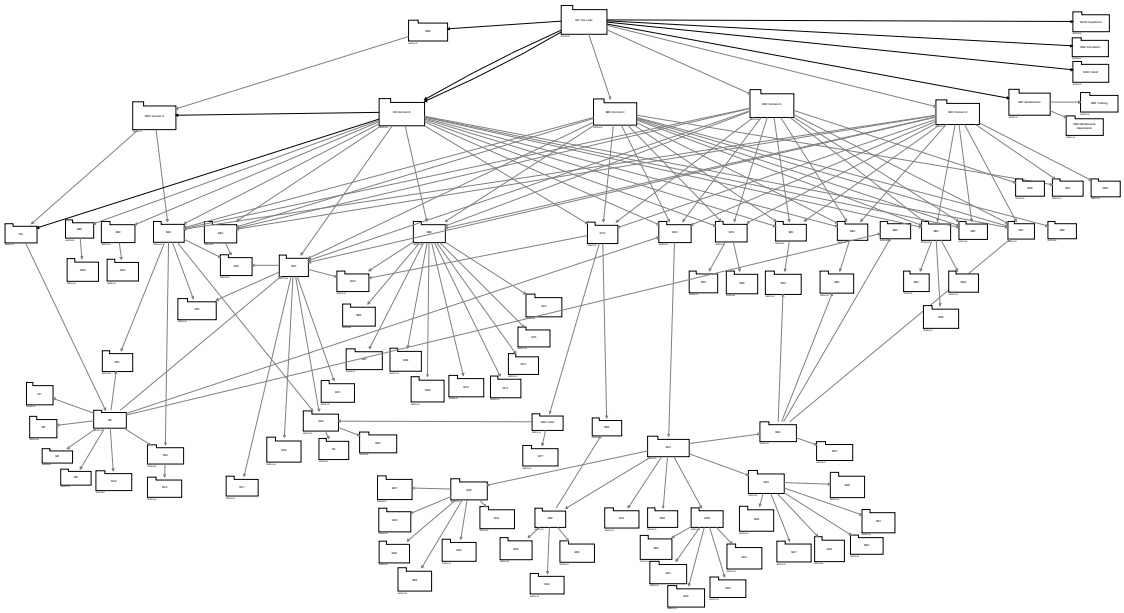

Figure 1: Example modular safety architecture
Note the diagram is illustrative, intended only to highlight the complexity of the assurance interactions that can be present in a typical safety critical system.

The primary links between *modules* are *dependency-claim* relationships, the *dependency* of one *module* (for example the availability of power > x%) is linked to the *claim* of another module (i.e. the power module claim for availability > x%). Note there may be many links between the same *modules* and these can be rolled into a single link to avoid over-complication. However, each link also has *context* and part of the *module* linking process is to ensure that the links are valid in the *context* relevant to both modules. For example, if the parent *module* requires power availability in extreme temperatures, this is not going to be met by a power supply that only works at room temperatures. From a safety perspective though, the *context* is often more comprehensive and can include, inter alia:

[9] The elements of the safety argument within a module that should be visible to other relevant parties within the TAS. Internal elements of the safety argument e.g. how a claim is substantiated need not be visible, nor is it necessarily of interest to other parties. These form the private part of the module.

- The level of safety assurance to which the claim is demonstrated, including risk criteria applied, assurance levels, etc.
- The scope of the system or service addressed by the module
- Any legislation or standards that the system or service meets
- Any caveats that amend the pure meaning of the claim such as assumptions about maintenance, limitations on use, or constraints on installation or adaptation.

Once valid links are established the parent and child *modules* retain responsibility of maintaining the interface AND addressing the implications of any *context*. For example *assumptions* and *limitations* inherited from a child *module* should be validated in the parent *module*; where this is not possible, they must be captured as *context* to the parent *module* claims. Whichever way the responsibility is split between *module* owners, someone needs to ensure that the links are valid in the given context, this latter role is referred to herein as the *argument architect*.

> Non-voice systems for controller-pilot communications (obviously) involve changes in (and affect) multiple domains (aircraft manufacturer, aircraft operator, ANSP). The European ATN system was developed largely from the ATM perspective with insufficient consideration of the aircraft (cockpit) end of the system. Development did not adequately consider: the need for certification of the airborne system, the human factors issues in the cockpit, the integration with the existing FANS system (providing similar service, used in Pacific and North Atlantic), the need for training of operators in use of the system. Furthermore, the system was novel with no existing AMC. An ongoing concern is that there may be pressure to increase the use of the data communications system (currently limited to cruise phase, non-time critical messaging): the safety implications of this change of use would need careful consideration. The logical argument approach provides a framework to consider the total impact of introduction of this system. It also allows a performance based approach (where the specifications required for a compliance based approach are not available), while allowing a compliance-based approach where the existing AMC material remains sufficient. The approach would also identify and define the interfaces between domains and stakeholders, allowing more efficient management of them. However, success of the approach does rely on co-operation between each domain (and geographic area?) within the TAS, and on identification of an "argument architect" to own and maintain the certification argument throughout the lifetime of the system.

2.2.2 How to decide on module boundaries

Success depends on appropriate modularisation of the argument by the *argument architect* into modules which support both construction of the argument and its reuse. Systems theory dictates that successful modularisation depends on modules being loosely coupled and highly cohesive. However, application of a modular structure onto a system with diverse *domains* also requires consideration of how and whether to align *module* boundaries with *domain* boundaries. Modularisation boundaries are usually aligned with one or more of the following as these align with extant structures.

- divisions of responsibility;
- organisational structure;
- system architecture;
- phases of the lifecycle.

Modularisation can also be used to contain volatile parts of the argument in an attempt to minimise the effect of this volatility on other parts of the argument – obviously the key to this success is the ability to define a stable interface for the module. Management of change is discussed further in section 2.2.5.

Modularisation of the *TAS* is discussed further in section 3.3.

2.2.3 Definition of module interfaces

In a modular *argument architecture*, the detail of the *argument* can be private, i.e. "hidden" within the *module*, although obviously available to review to confirm that *module claims* are demonstrable. The interface of the *module* should be public, and defines a number of attributes (numbered for ease of later reference):

1. *Claims* made by the module – i.e. those which the *module* provides the *argument* to support
2. *Evidence* presented by the *module*.
3. *Context* defined in the *module*, including scope and boundary, applicability, system or service capability, etc.
4. *Claim* caveats – *assumptions*, *limitations*, safety issues (e.g. non-conformities) and *constraints*
5. *Dependency - claims* identified within the *module*, but which another module provides the argument to support
6. References to *evidence* presented in other *modules* which is required to support the current *module*.
7. References to *context* defined in other *modules* which forms part of the *context* for the current *module*.

Clear definition of the interfaces between modules is critical to the success of the modularisation; some suggested layouts for interface definitions are provided in [16]. (It should be noted that [16] identifies 7 different attributes at the module interface: two of these have been subsumed into item 5 above for simplicity within the treatment here.)

Definition of an interface then makes it possible to establish *assurance contracts*[10] between *modules*; this approach is particularly useful when *modules* are being developed by different organisations as it allows a clear definition of the responsibilities of each. *Assurance contracts* can provide a basis for managing the validity of links between *modules* and holding a common single link point for child *modules* with multiple parent *modules*. Such *assurance contracts* may, and in many cases do, exist already and thus should be

[10] Within the research papers for Modular Safety Arguments the term "contract" is used to denote the formal arrangement between two or more modules. For the purpose of ASCOS these are referred to as *assurance contracts* to avoid potential confusion with commercial terminology. During application of the proposed approach consideration will be given to alternative names that best match the emerging nature of these contracts.

aligned with any relevant existing standards, for example those for Mode S. However, existing standards may not individually or collectively capture all of the necessary elements of an *assurance contract*.

> A study in the mid-1990s identified that in 70% of accidents involving airplane systems, the original design assumptions were inadequate for the situation existing at the time of the accident due to changes in: the aviation system, airplane operational usage, personnel demographics, evolving infrastructure or other considerations. The continuing complexity and diversity of changes, including the changes in underlying technology can only serve to exacerbate the situation. This shows the criticality of documenting context and assumptions within a certification argument, and how it is critical to continually monitor these items through the lifetime of the system (Cl 5 of the generic argument presented in section 3.2).

2.2.4 Verifying the overall *argument architecture*

It is also essential, once the *argument architecture* has been developed, for the *argument architect* to verify that the resultant *argument architecture* is complete, consistent and correct, using the following steps:

1. *Claim* matching: to ensure that any *claims* requiring support from other *modules* (item 4 above) are fully made (item 1) in another *module* of the *argument*.
2. Consistency checks: to ensure that *evidence* (item 2) and *context* (item 3) are consistent across all *modules* of the *argument*. This is probably the most difficult part of this process as it involves reviewing the whole body of *assumptions* and *evidence* to identify any conflicts or inconsistencies.
3. Cross reference checks: to ensure that cross references to *evidence* and *context* (items 6-7) refer to items which exist and which support the *argument* as intended.

2.2.5 Management of change

The purpose of modularisation of the *argument* is to split the *argument* into chunks of manageable size so that they can be developed separately. The benefits of this approach are most apparent when an *argument* needs to be changed - this could be for one of a number of reasons, including:

- change within the system;
- part of the *argument* is found to be incorrect;
- inability to provide the *evidence* envisaged when the *argument* was constructed;
- counter evidence produced during in service monitoring.

If the modularisation has been carefully chosen, it should be possible to limit the impact of the change to a single *module*, or a small number of *modules*. Although it will still be necessary to repeat the verification step (see section 2.2.4), this should also be simpler, as only the items which have changed (and their effect on other items) need to be revisited.

In addition to careful choice of *module* boundaries (see section 2.2.2), careful application of the following principles can reduce the degree to which change propagates outside affected modules – this therefore assists in minimising the impact of changes.

1. **Avoid unnecessary restriction of *context***: When defining *context* make the definition as broad as possible: for example, if different *modules* make differing assumptions about operating temperature (e.g. module A assumes 10-20°C and module B assumes 20-30°C) the *context* is not consistent. However, it may be possible to extend these ranges without adverse effect on the *argument*. If this is done at the outset, it prevents inconsistencies when *modules* are combined.
2. **State *dependencies* as limits rather than objectives**: Define the *claim* based on the minimum level of support which is sufficient to make the *argument*, rather than on the level of support which would ideally be available.
3. **State *dependencies* as ends rather than means**: Define the *claim* based on what needs to be demonstrated, rather than how it should be demonstrated. This gives maximum flexibility to the module making the *argument*, with the potential side effect of making that *claim* more easily reusable in other parts of the *argument*.

2.3 Representing the *Argument*

An *argument* may be presented in a variety of implicit or explicit forms often purely textual or compliance based. However, explicit use of a graphical notation with a formal syntax encourages greater consideration of the logical construction of an *argument* and allows it to be more readily understood and thus challenged where it is incomplete, incorrect or invalid. (This is an application of the English saying "A picture is worth a thousand words.") The Goal Structuring Notation (GSN) is an example of a graphical argument notation and was developed for this specific purpose. It has been successfully applied in many safety critical domains, including avionics, aviation, nuclear and rail.

GSN is now defined in a published standard [14] and supported by multiple research papers and presentations, including those listed in section 2.2. It is also described in the EUROCONTROL Safety Case Development Manual (SCDM) [15] and in a UK CAA guidance document on production of safety cases [23]. It should be noted that these documents do not specifically recommend any particular graphical approach. In addition the safety argument modularisation approach does not require the use of any particular argument notation.

GSN is chosen for this work package over other graphical notations as:

- it is formally defined in a community standard;
- it is flexible in its use, with a developed notation for modularisation of arguments;
- there is tool support available for verification of arguments and the modular safety argument notations;
- it supports the definition of template arguments that can be applied to similar systems or services;

- it is already used within the civil aviation industry[11] and is covered in industry publications as described above.

A summary of the GSN notation is presented in Appendix G.

2.4 Fallacious Arguments

As with all forms of scientific reasoning, mistakes can be made or poor reasoning can be used in the construction of arguments, resulting in fallacious arguments. Arguments can be fallacious whether or not the conclusions are true. Fallacious arguments fall into two categories:

- A *formal fallacy* is a pattern of reasoning that is always wrong. This is due to a flaw in the logical structure of the argument which renders the argument invalid.
- An *informal fallacy* is an argument whose stated premises fail to support its proposed conclusion.

The OPENCOSS deliverable D4.1 [4] (section 3.3.3) contains a summary taxonomy (first published in [36], and listed below) of common mistakes made, which can lead to fallacious arguments. However, arguments may not be incorrect or inadequate just because they exhibit the characteristics of these fallacies. See [52] and [54] for examples of some of the issues with circular arguments and appeals to expert judgement.

- Circular reasoning occurs when an argument is structured so that it reasserts its claim as a premise or defines a key term in a way that makes its claim trivially true.
- Diversionary arguments contain excessive amounts of irrelevant material that could distract a reader from a weakly supported claim.
- Fallacious appeals invoke irrelevant authorities, concepts, or comparisons as evidence.
- Mathematical fallacies describe common pitfalls in probabilistic and statistical inferences.
- Unsupported assertions are claims stated without evidence.
- Anecdotal arguments show that their claims hold in some circumstances but fail to generalize their validity.
- Omission of key evidence occurs when an otherwise complete argument omits evidence that is necessary to establish its validity.
- Linguistic fallacies concern the use of misleading language that might lead the reader to an unwarranted conclusion. These fallacies may appear in any informal argument.

Another significant issue in the application of logical thinking is the notion of “confirmation bias”, essentially the tendency of people to favour information that confirms their beliefs or in this case positive claims about their system. Whilst clearly common to all forms of reasoning and scientific enquiry it is particularly prevalent in the absence of any clear rules, structure and guidance (e.g. that is provided by comprehensive certification specifications). However, even in this latter example case the belief that a system of certification is adequate and effective can long outlast evidence that shows otherwise. To this end it is important that arguments are

[11] In particular, in safety cases made by civil ANSPs.

developed in a scientific manner taking into account a balanced view of all relevant evidence (both confirmational and falsifying), including an active search for counter evidence in relation to any claims. Note that in a scientific approach, a hypothesis is stated and then the main part of the research is aimed at rejecting the hypothesis. The same approach should also be considered in the substantiation of claims.

A reading list in relation to fallacious arguments is presented in Appendix F.

3 Logical Argument Approach to Aviation Certification

3.1 Outline of Application to Aviation Certification

The central proposal of the proposed certification approach is to develop a logical argument for changes[12] to the *total aviation system (TAS)*: the top level *claim* of the *argument* would be that the (defined) change[13] to the *TAS* is acceptably safe. As explained in Appendix B, the approach considers the whole system of people, processes and equipment.

A prerequisite for making any safety argument is an adequate definition of the change which is being made. This must specify the operational and functional requirements of the complete change and define a high level architecture which then allows allocation of requirements (including safety requirements) between the components of the system. (Naturally, this definition will initially be at a high level and the level of detail will be developed progressively, in line with the stages described in section 4.)

The critical first step in constructing an *argument* is the precise definition of the top level *claim* and the supporting *context*, *assumptions* and *justifications*. Some examples of these elements are as follows:

- *claim*: The dual arrival streams concept at airport X has been specified to be acceptably safe.
- *context*: Acceptably safe is defined as satisfying the safety criteria such that (1) risk shall be no greater than tolerable and (2) risk shall be reduced ALARP.
- *context*: The concept is specified and assessed for all normal, abnormal and degraded modes and conditions.
- *assumption*: The existing operations at airport X are acceptably safe.

Accurate and complete identification of all these supporting elements is critical as it guides (and may constrain) the approach taken to make the argument.

The top level *claim* is then decomposed into *sub-claims* where satisfaction of the *sub-claims* is sufficient to demonstrate that the higher level *claim* is satisfied. This process is repeated to derive successively lower levels of *claim* (with accompanying *strategy*, *context*, *assumptions* and *justifications* as necessary) until each individual *claim* can be directly supported. The decomposition will be aligned to the division of responsibility with the *TAS* and limited to that necessary to support definition of the interface between the *TAS domains*,

[12] Section 3.1.2 discusses the triggers for considering a change as something which requires an argument to be constructed.

[13] The ideal approach would be to demonstrate that the overall *TAS* is acceptably safe in the context of each change introduced. However, development of such an *argument* from the outset would be an enormous and complex task, and unlikely to be achievable. The approach proposed here does require the impact on all parts of the *TAS* to be identified and managed. Furthermore, as *arguments* are developed for successive changes, these will build towards an *argument* for the safety of the *TAS*. Indeed, it will be necessary to co-ordinate and maintain these *arguments* to ensure that their *assumptions*, *constraints* and *limitations* remain compatible with each other. It is noted that, although an earlier study [56] recommended construction of a Whole Airspace ATM System Safety Case, this recommendation does not appear to have been progressed.

and to dovetail with the existing certification approaches and specifications within each *domain*. Where existing standards are insufficient, this will support definition of new specifications to support the introduction of novel technology or concepts. Development of such standards streamlines future applications of the same or similar technology or concepts and provides a forum for establishment of interfaces between components of the system. Where neither of these approaches is available, *claims* may need to be developed to a more detailed level, but it is intended that this would be the exception.

There is a generic *argument* already widely used within ATM which could be adapted for use as the highest level of a generic *argument* for the safety of any change to the *TAS*. Originally developed within the EUROCONTROL Safety Case Development Manual (SCDM) [15] and subsequently extended and enhanced by the SESAR research programme (see [37]) this generic *argument* addresses all stages of the development lifecycle, from concept through to maintenance in continued operation. This *argument* is considered to be a reasonable starting point for consideration during the case studies, but the expectation is that this template will evolve further as a result of its use across a greater part of the *TAS* than just ATM.

The natural development of the *argument* is to develop each "leg" (broadly representing a lifecycle stage) in sequence: thus the approach can support *progressive certification*, where early "legs" of the *argument* can be approved to give confidence in the approach before the whole *argument* is complete. The *argument* is largely independent of any given system or service change and is suitable for novel and well established systems alike. In the latter case it is possible to rapidly address some of the *arguments* by reference to existing documentation such as equipment specifications for example. This generic *argument* is presented in more detail in section 3.2.

Within each *domain* affected by the change, there will be a standard approach (or approaches) which are taken to achieve *certification*. These standard approaches form part of the *context* for the *argument*. During the development of the *argument* the engineer(s) involved will decide whether this usual certification approach can be used to meet the *claims* of the *argument* for the current change. In other words, the engineer will ask the question: "Will this approach deliver the evidence needed to demonstrate that this (part of) the change to the *TAS* will be acceptably safe?" An advantage of the logical *argument* approach is that it provides flexibility to adopt an alternative approach if the usual approach is not suitable.

The decision of how to decompose the *argument* is supported by the provision of template *argument* structures for different (standard) approaches, allowing the engineer to select and develop the template which is most appropriate for the change. Some template *arguments* are provided in Appendix D. It is intended that further templates will be provided in the final version of the proposed certification approach following completion of the ASCOS case studies. It should also be noted that the top level of a *performance-based* certification approach is provided in section 3.2.

The flexibility of the approach allows approaches from across the aviation industry (and potentially from outside it) to be applied as appropriate. Where an established approach is used, it is only necessary to develop the *argument* to a level sufficient to demonstrate that the approach allows the high level *claims* of the *argument* to be met. Examples of cross-domain fertilisation are discussed in section 3.4.4.

In order to make a clear and unambiguous *argument*, it is necessary to establish a common language to express the safety *argument* concepts. This language must be used for safety *arguments* across all *domains* within the *TAS*, to ensure that the concepts are understood in the same way by all who are involved. (This was one of the recommendations arising from previous ASCOS work.) Building on the work of the OPENCOSS programme (see Appendix E.1.1), we have developed a lexicon of terms (see Appendix A) for use in developing these logical *arguments*.

The advantage of this approach is illustrated by two examples which were highlighted in interviews with certification experts:

- **Composite airframe material**: When attempting to certify an airframe containing a new composite material it became apparent that the relevant part of CS25 [30] was based upon conventional metallic construction crash performance requirements (e.g. strength, fatigue and dynamic crash response). Taking a logical argument approach here allows us to take a step back and ask the question "What does the applicant need to do in order to show that the composite airframe will be acceptably safe?" The initial answer might be that the composite material must deliver the same performance as the metallic materials which it is replacing. Thus the argument approach includes determining the levels of performance delivered by the metallic materials, and then demonstrating that the composite materials do indeed meet these levels.
- **Electronic flight bag (EFB)**: EFB equipment may be provided on new aircraft and certified as on-board equipment, but this does not mean that it is necessarily certified for use. This is a novel technology so there are no established certification requirements, so it is necessary to fall back on a "first principles" argument approach. In the context of the function for which the EFB is to be used it is then necessary to consider (a) fallback in event of failure and (b) continued safety in operation – these are parts of the standard argument presented in section 3.2. Developing these parts of the argument reveals the concerns which need to be addressed, in particular ensuring that both the EFB and any fallback equipment remain up to date. (A fallback is no use if, in the event it is needed, the charts within it are out of date and therefore unusable.)

Previous ASCOS work identified *bottlenecks* and *shortcomings* in the existing approach to certification. Section 3.4.3 describes how these are addressed within the context of a logical *argument* approach.

3.1.1 Previous uses of logical *argument* in aviation

The logical *argument* approach has already been successfully applied to achieve *certification* for novel concepts in certain parts of the *TAS*, providing a degree of confidence of its suitability for use in the *certification* of further novel concepts proposed for introduction in the European aviation industry. Furthermore, the preparation of a safety case for functional airspace blocks is required in EC legislation ([55] Article 9a). The suitability of the approach will be proven in the forthcoming activities of the ASCOS programme.

Past applications of the approach are numerous but include:

- the operation of military Unmanned Aerial Vehicles (UAVs) in non-segregated airspace [25];
- Reduced Vertical Separation Minima (RVSM) [26];
- the development of the Point Merge operational concept [49];
- the introduction of ACAS into European airspace [50].

It should be noted that the RVSM safety case was an early application of the approach and has been subject to extensive review and criticism in the safety community. The flaws identified emphasise the importance of:

- discipline in argument development to avoid unnecessary complexity; and
- rigorous review of logical *arguments* to ensure that they are correct and consistent.

It should be noted that just because an argument contains flaws, it does not render the overall argument untrue. This issue is addressed further in section 2.4.

The approach is also embodied in UK CAA safety requirements publications, including:

- CAP670: Air Traffic Services Safety Requirements [24]
- CAP760: Guidance on the Conduct of Hazard Identification, Risk Assessment and the Production of Safety Cases [23]

3.1.2 Triggers for change

A change to the *TAS* may be triggered by one of the following:

- a business need – e.g. to improve the capacity of the *TAS*;
- a change to the environment within which the TAS operates; (Of course, a change to one part of the *TAS* may drive a consequent change to another part of the *TAS* but these are logically considered as part of the same change, which would be driven by one of the triggers listed here.)
- continuous safety monitoring identifying a need to improve performance to ensure that safety targets are met.

However, the proposed approach presented in this document would only be triggered if the change requires explicit acceptance by the competent authority (or authorities) affected by the change. Such acceptance would only be required where the change is outside the established procedures and processes for the system. For example, a planned maintenance activity, including like for like equipment replacement, will usually be covered under the existing safety case and processes and would not lead to application of this approach.

During the refinement, following the case studies, further consideration will be given to defining triggers for application of this certification approach and to its integration into the existing certification processes.

3.2 A Generic Argument

This section presents the generic logical argument (see Figure 2) which forms the core of the proposed certification approach. This argument is not mandated by standards, but is commonly adopted for development of safety cases in the ATM *domain*. It is presented in the EUROCONTROL document Safety Assessment Made Easier [31] and also referenced in the SESAR research programme [37]. This argument has been chosen as a suitable starting point for development in the ASCOS case studies. Another template argument (which may be more appropriate when considering an organisation) is presented in Appendix D. Experience gained during the case studies will be used to define refinements of these templates, as well as any further templates needed, to facilitate application of the approach to changes across the whole scope of the *TAS*.

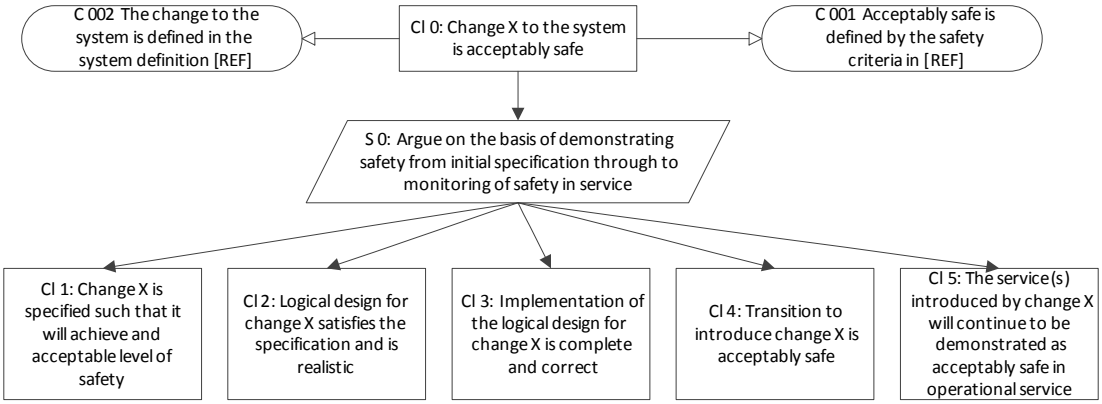

Figure 2: Generic Logical Argument

The *argument* starts with the top level *claim* (Cl 0: "Change X to the system is acceptably safe"). Before we start to decompose the *claim* we need to define the context of the change, which usually includes:

- precise definition of the change being made, including the reason(s) for making the change – where this is replacement of an existing system, this should include any changes in functionality between old and new systems
- definition of the term "acceptably safe", through definition of safety acceptance criteria
- *assumptions* about the environment (including the surrounding system) within which the change is being made
- applicable regulations
- identification of novel features or functions which may be outside the current understanding of those within the system.

Often, changes are designed in the context of equipment which has already been developed, although it is recognised that this is not ideal, as it does not provide the opportunity to drive development from the actual requirements of the change. In these cases, it is necessary to carefully consider the assurance level required for the equipment to ensure that a sufficiently robust argument for safety can be made.

Context should be defined at the highest level at which it is relevant; this can result in *context* being refined as the *argument* is decomposed.

At the next level the *argument* is broken down into *claims* which address the different stages of the development lifecycle[14]. The development of these claims is further addressed in steps of the process described in section 4; this includes a discussion of the processes used and the outputs of each stage of claim development.

- **Cl 1: Change X is specified such that it will achieve an acceptable level of safety**: This *claim* focuses on *what* is being changed (e.g. introduction of a new concept or service) without considering the details of how the change is implemented. At this stage, the change is considered at the functional specification level, in the context of high level functions, operational behaviour, modes of operation and scenario analysis. (However, even at this level, the change should be partitioned into the different *domains* within the *TAS* to facilitate initial development of the argument.) In an ATM argument, for example, this *claim* is made at the operational level, considering the paths which the aircraft take through the airspace, without considering the tasks or equipment employed to guide them to these paths. This *claim* includes the performance of the change as specified (including consideration of all normal, abnormal, degraded and emergency conditions) in the absence of failure.
- **Cl 2: Logical design for change X satisfies the specification and is realistic**: This claim demonstrates that the logical design[15] of the change has the functionality and behavioural and performance attributes necessary to satisfy the specification considered in Cl 1. This claim considers all normal, abnormal, degraded and emergency conditions of the operational environment. In addition, this claim considers all the possible hazardous failure modes of the logical design and sets mitigations and assurance requirements such that the system is acceptably safe in the presence of these failures.
- **Cl 3: Implementation of the logical design for change X is complete and correct**: This *claim* demonstrates that the physical implementation[16] of the change correctly implements the design. As well as directly ensuring that all the requirements are met, this part of the argument also assesses the design to ensure that any inadvertent adverse safety properties are identified and (where appropriate) mitigated. It is to support this claim that detailed assessments of the failure modes of the equipment, people and operations are made.
- **Cl 4: The transition to introduce change X is acceptably safe**: This *claim* is concerned with preparing the system (equipment, people and procedures) for bringing it into operational service. It also includes the question of how the system can be brought into service without adversely affecting the safety of the existing on-going operations during the period of the transition from the current operations to the new situation.

[14] The argument is mapped to the E-OCVM lifecycle in section 3.2.1.
[15] In this context, logical design is a high-level architectural representation, independent from the physical implementation. As such it considers the functions provided by the system elements (i.e. human roles and tasks and machine-based functions), but not the equipment, personnel or procedures which provide these functions.
[16] Physical implementation includes the details of equipment (hardware, software and data), people (flight crew, controllers and maintainers), operation and maintenance procedures, training and sectorisation.

- **Cl 5: The service(s) introduced by change X will continue to be demonstrated as acceptably safe in operational service**: This *claim* is concerned with (a) ensuring that the *a priori* safety assessment (made in arguments 1 – 3) is supported by in service evidence (and addressing any deviations of the actual system from the predicted performance) and (b) with ensuring that any changes[17] to the system or its environment are correctly monitored (and that any corrective actions needed are implemented). It is here that complete and accurate identification of the relationship between the part of the system being changed and the rest of the *TAS* and the *external environment* is critical: this is necessary so that the correct items in the *TAS* and the *external environment* can be monitored and so that corrective action can be taken where necessary.

This *argument* structure emphasises the need to ensure both that the change is safe in the absence of failure and in the case of failures or errors. In other words ensuring that, if the changed system functions as designed, it will be acceptably safe. This is a crucial step which can be overlooked in a purely failure-based approach. The need for this part of the argument (sometimes termed the "success case") arises because there are intrinsic hazards within the aviation system (e.g. conflict between aircraft trajectories or controlled flight towards terrain) which systems such as ATM systems are introduced to prevent. Too much focus on the analysis of failures within the system may miss the fact that the design (when functioning correctly) does not meet the safety objectives.

As discussed above, decomposition of the claims is undertaken (only) to the level where claims can be directly aupported, although the approach allows for decomposition to lower levels where insufficient support is available from other approaches.

Decomposition of *claims* can be supported by *strategies* which explain the approach taken in the decomposition. Use of *strategies* helps to explain the *argument* and assists reasoning as to the completeness and correctness of the *argument*.

When the arguments presented above are decomposed further, each includes direct arguments – i.e. those supported by *direct evidence* supporting the claim – and backing arguments – i.e. those arguing that competent processes and people were used to produce the *direct evidence*.

For the purposes of this presentation, the *argument* will not be decomposed further, although more detailed arguments will be developed during the case studies, which will illustrate the usual structure and approach taken in each leg of the argument.

3.2.1 Mapping to the E-OCVM Lifecycle

Table 1 shows the rough mapping to the E-OCVM (version 3.0) lifecycle adopted by ASCOS WP3. The lifecycle is presented in full in [9].

[17] Changes to the system in operation may be through degradation of the equipment or through intentional changes following the initial introduction; changes to the operational environment would include changes in the way in which the airspace is used.

Argument Leg	*E-OCVM Lifecycle Stage*
1: Specification	V0 (System Needs); V1 (Scope); V2 (Feasibility)
2: Design	V3 (Preindustrial development and integration)
3: Implementation	V4 (Industrialisation); V5 Deployment
4: Transition	V5 (Deployment); V7 (Decommissioning)
5: Operation	V6 (Operations); V7 (Decommissioning)

Table 1: Mapping the generic argument to the E-OCVM lifecycle

Notes:

- V5: Deployment (as defined in [9]) spans the implementation of a product or concept to a specific site or location and the introduction into operational service.
- Decommissioning of existing concepts, systems or equipment is addressed in the Transition leg (Cl 4) of the argument for introduction of the replacement arrangements.
- Decommissioning may also be addressed under the Operation leg (Cl 5) of the argument, where decommissioning is undertaken under the management arrangements established for the introduction of the change.

3.2.2 Links to WP2: Continuous Safety Monitoring

The main aim of WP2 is to develop a methodology and the supporting tools for multi-stakeholder Continuous Safety Monitoring (CSM), using a baseline risk picture for all parts of the *TAS*.

The CSM process will be useful in both *a priori*[18] and *a posteriori*[19] risk assessments.

The data generated by the CSM support *a priori* risk assessments by providing (predictive) quantifications of the probability and / or frequency of occurrence of events within the system, supporting the overall estimation of risk required to demonstrate that the system is capable of meeting the safety requirements derived during development of the safety argument. This includes using the data to support any quantified assumptions about how the system will behave.

The CSM data will have its own context – the operational environment from which it was gathered. A number of parameters define this environment, including the numbers and types of aircraft, the types of navigational equipment in use, the separation implemented by ATM and the operational processes and procedures used. When the CSM data is used to support an argument, this context must also be adopted. Where the change being implemented changes some of these parameters, the potential effect on the data must be assessed.

[18] An *a priori* risk assessment is one which is undertaken before the implementation of the change – i.e. one used to support Cl 1 – Cl 3.
[19] An *a posteriori* risk assessment is one which is undertaken retrospectively using data from experience of the change while in operation – i.e. the assessments undertaken to provide ongoing safety assessment as described in Cl 5.

The CSM process supports *a posteriori* risk assessments by establishing the framework for collecting data. As part of Cl 5 for a specific change, specific metrics (SPIs) will be identified to monitor the safety in service of the change. Some of the required metrics will already be defined within the overall CSM scheme defined in WP2. It is necessary for the CSM process to be sufficiently flexible to allow additional metrics to be added where necessary to support the safety monitoring required to fulfil Cl 5 for a specific change.

3.2.3 Links to WP3: Safety Risk Management

The main aim of WP3 is to develop a *total aviation system* safety assessment methodology, with supporting safety based design systems and tools, for handling of current, emerging and future risks. This is to be achieved by representing current and future risks in accident and accident avoidance scenarios in such a way that it can be used in the certification process.

The methodology developed in WP3 supports the analysis required to support the argument for the functional specification (Cl 1), the logical design (Cl 2) and for the implementation (Cl 3). This includes both the safety system when functioning as designed and the analysis of failures and failure modes.

Where WP3 identifies prevention models for the risks, these prevention models should be considered in the argument, as part of the demonstration that the application of these models delivers a system which meets its safety criteria.

The logical argument approach does not impose any particular failure analysis approach, but does provide a framework for integrating the approaches taken in different *domains*.

3.2.4 Decomposition of the Argument

As noted above, the *argument* needs to be decomposed into sub*claims* until a level is reached where the *claims* can be directly supported by *evidence*.

It is difficult to give detailed guidance on decomposition of the *argument* because, by its nature, this is a creative exercise. However, the following principles will facilitate the alignment between the *argument* and the supporting *evidence*, either generated through application of existing standards or through the supporting tools developed in WP2 and WP3.

1. The *argument* should make the link between the certification requirements, as expressed in the relevant regulations, and the *evidence* which would be produced by following the relevant guidance (e.g. standards, AMCs) in use within the domain. The development of the *argument* should be specific about the *evidence* required and why it is required, in order to support the exercise of reviewing the argument to determine whether the *claims* in the *argument* are satisfied by the *evidence*. Where development of the *argument* leads to requirements for *evidence* which would not be produced by following the usual processes within the domain, this highlights the fact that additional approaches need to be defined and followed.

2. *Claims* should only be decomposed to the level where they can be directly supported. Ideally this support would be through application of existing certification approaches, or through the development of new standards to support the introduction of novel technology or concepts. Development of such standards streamlines future applications of the same or similar technology or concepts and provides a forum for establishment of interfaces between components of the system. Where neither of these approaches is available, *claims* may need to be developed to a more detailed level, but it is intended that this would be the exception.
3. The *argument* should demonstrate that each of the relevant risks identified within the risk model (as developed in WP3) has been addressed. This could be through explicit enumeration of each individual risk within the argument or it could be through demonstrating that a process has been followed which ensure that all the risks have been addressed.
4. Similarly, the logical argument should take into account the barriers (to risk propagation) identified within the risk model and should demonstrate that the failures of those barriers have been properly considered and addressed.
5. Consideration should be given to creating guidelines on the rigour of *evidence* required to support each *claim* made by the *argument*. This should take into account the types of evidence available and the diversity between these types of evidence. Where only one or two sources of evidence are available and / or where these evidence come from similar sources, a much higher degree of confidence is required in each piece of evidence than where more (or more diverse) different sources are available. In some *domains* and /or types of *argument*, it may be possible to develop a metric for the degree of rigour required.

3.3 Modularisation of the Generic Argument

As discussed in section 2.2, modularisation allows subdivision of the *argument* into well-defined *modules*, with well-defined *interfaces* so that these *modules* can be developed separately from one another in confidence that the final result will be a consistent and correct overall *argument*.

When engineering complex systems, any modularisation requires a system architect to design and ensure the integration of the resultant modules. Similarly the *argument architect* is responsible for ensuring that the argument modules are correctly bounded and interfaced to other modules. However, when considering the number of organisations involved in the *TAS* and their disparate roles, it is not easy to identify who should be the *argument architect*. This in part explains why *shortcomings* are reported in the interfacing of *domains*; sometimes integration is supervised by the competent authority or even ignored altogether. Whilst the competent authority is in a position to oversee the architect role it would be inappropriate to task regulators with engineering the integration. At this stage it is proposed that integration remains the responsibility of individual organisations to:

- Ensure the *module* definition is complete and correct with respect to the *evidence* it contains.
- Ensure that all *assurance contracts* are agreed with the relevant interfacing organisations.

It then remains the responsibility of the competent authority/ies to provide oversight of the above arrangements. Due to the way in which such *arguments* span multiple domains, there may not be a single authority competent to endorse the overall *argument*. As a result, it is necessary during the initial planning of the certification approach to clearly define the parts of the argument which require endorsement by each competent authority.

The remainder of this section presents a possible *argument architecture* for the safe operation of Electronic Flight Bag (EFB) technology. This is an extension of the scenario originally outlined in section 3.1. This example is presented purely to illustrate modularisation; in a real application it would be necessary to consider the intended function of the EFB in detail.

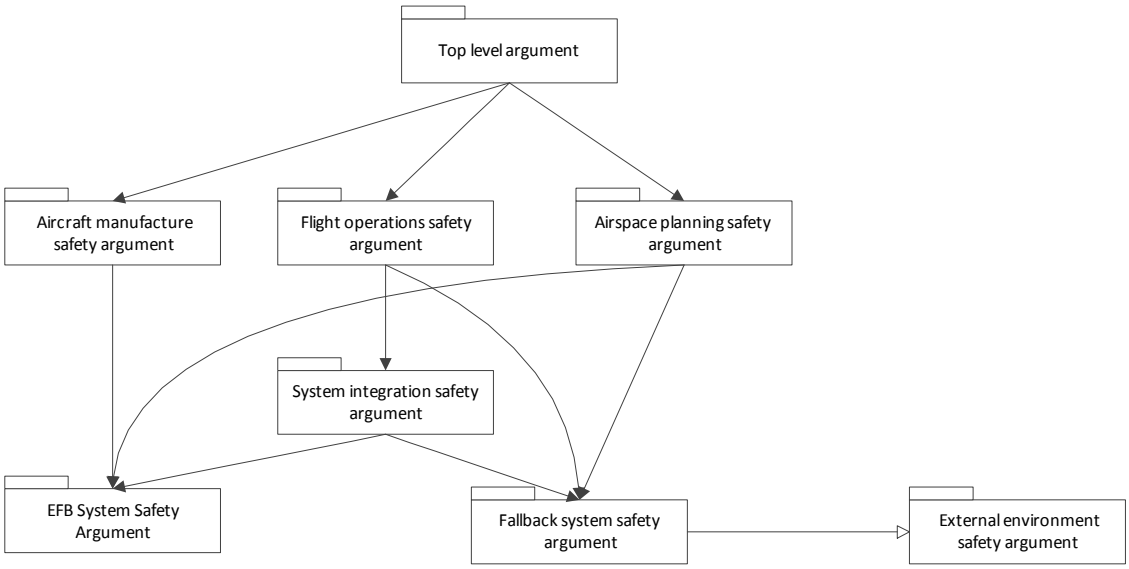

Figure 3: Modular Safety Argument Architecture for Operation of Electronic Flight Bag (EFB)

In this example, *modules* are used for a number of purposes:

- as a "wrapper" around existing safety case material, identifying the *claims*, *context*, *constraints*, *limitations* and *assumptions* made in the safety case, to allow these to be integrated into the rest of the *argument*;
- as an interface between the different *certification* approaches in the different *domains* (e.g. between aircraft operator, aircraft manufacture and airspace planning);
- as a container for issues relating to integration of the overall system;
- as an aid to developing the safety requirements for individual parts of the solution, by containing the *argument* relating to different products in different safety case *modules*.

Another potential use for modularisation, which is not shown here, but which could conceivably become part of this argument as it is developed, is to facilitate separation of *direct evidence* from *backing evidence*. This can be particularly useful where the same processes are used to generate evidence in different parts of the

argument: rather than justifying these processes multiple times, this *justification* can be captured once in a separate *module* and then invoked as *context* within the direct part of the *argument* where necessary. For more on this approach, refer to [27].

It can be seen that, in this case, the high level modules are more abstract, while the lower level modules deal with more concrete parts of the system.

It should be noted that although the introduction of the EFB may not require changes to the services provided by the ANSP, this *domain* is still represented in the *argument*, because *assumptions* are made about the services provided. Similarly, a *module* has been defined to represent the *external environment*, to capture the *assumptions* made by the *argument* about this environment.

The *argument architecture* shown here is not a substitute for a full representation of the *argument*: this diagram only shows how the various "chunks" of the *argument* fit together, and would need to be accompanied by the full definition of the *argument* within each *module*, as well as verification (as described in section 2.2.4) that the modules, when composed together, do form a complete and consistent argument.

3.4 Application to the most promising options for certification process adaptation

Previous work examined the current practice in the various *domains* of the *total aviation system* and identified and evaluated eight possible options for improvement of the certification process [1]; the benefits of each option were then evaluated against a number of criteria.

This section explains how each option is realised within the proposed approach as listed below:

- Section 3.4.1: Option 2: Change between *performance-based* and *compliance-based* or vice versa
- Section 3.4.2: Option 6: *Proof of concept* approach
- Section 3.4.3: Option 7: Enforce existing rules and improve existing processes
- Section 3.4.4: Option 8: Cross-domain fertilisation

Where appropriate, template arguments have been developed, and are presented in Appendix D.

3.4.1 Option 2: Change between *performance-based* and *compliance-based* or vice versa

In general, a *compliance-based* approach is more suited to the application of an established technology for which a detailed specification already exists. Furthermore, before using such an approach, it is important to confirm that the specification to be used remains applicable within the *context* for the change to be made. (For example, the specification may have been designed specifically for operations in temperate climates. Operation in desert climates would introduce a completely different physical environment for the equipment, thus new environmental specifications for the equipment may be needed. However, the functional specification for the equipment (i.e. the functions it must deliver) may be unchanged.)

Conversely, a *performance-based*[20] approach is more suited to the application of a novel technology where the performance requirements are known, but where the novelty means that a detailed specification has not yet been developed. This approach requires the derivation of a detailed specification from the performance requirements: the assessment then demonstrates compliance of the design and implemented solution with this specification. If a specification is already available, even where the context of application is slightly different, it may be more effective to use a *compliance based* approach, but with focused areas of *performance-based* assessment to address the changes of context.

It should be noted that there is not always a binary distinction between *compliance-based* and *performance-based* approaches as, in practice, a combination of both may be used.

The generic argument presented in section 3.2 is a high level template for a *performance-based* argument; a template for a *compliance-based* approach is presented in Appendix D. These *arguments* demonstrate the differences between the approaches, and illustrate the type of activities and evidence which are needed for each approach. This information then allows the analyst to judge which approach is (more) feasible for the change under consideration.

Where the change requires a blend of approaches (as discussed above), an overall *argument* can be built by combining the *arguments* for the two approaches.

> Technology advances have made it possible to design an engine control system that automatically increases the thrust on the remaining engine(s) in case of engine failure. However, existing requirements require manual selection and back-up for this automatic feature, as previous automated systems were not reliable enough. Modern systems are so reliable that this manual back-up is no longer necessary, making it possible to remove unneeded components, thus removing failure modes and also increasing reliability. However, the authorities have not yet been able to develop an alternative requirement due to lack of resource. Adoption of a logical argument approach would open a way to implement this change without the need to first develop the alternative requirement and would ensure that all the possible impacts of such a change are adequately considered.

3.4.2 Option 6: *Proof of concept* approach

A *proof of concept* is a demonstration that a concept or product meets the requirements (or is capable of doing so), in order to support certification. This demonstration may use an early prototype before full development, or it may use a developed product in early deployment on multiple aircraft. The prototype may be an innovative, scaled-down version of the system or operation intended to be developed. The proof of concept approach has been developed by SESAR [17] [18].

[20] In this context, "performance-based" relates to demonstrating safety by achieving objectives, rather than specific performance parameters of the system.

Where the *proof of concept* involves introduction of uncertified equipment or processes into an operational environment, there are two types of argument which need to be considered[21]:

1. operational safety during *proof of concept* demonstration: this argument needs to show that the *TAS* remains acceptably safe while uncertified equipment or processes are under test through the *proof of concept* approach;
2. operational safety relying on evidence from *proof of concept* demonstration: in this case the *proof of concept* demonstration has a lesser impact on the argument – it is purely another means of generating evidence to support part of the argument.

Since the idea behind *proof of concept* is to introduce novel concepts, it is unlikely that specifications will exist through which the change could be certified and thus an approach not reliant on such specifications (e.g. an *argument* based approach) will be required. This option is therefore a good example of where the novel approach to certification will be useful.

An argument for operational safety during *proof of concept* demonstration could take the following form:

1. Determine the precise effects which the *proof of concept* has on the *TAS* while it is underway.
2. Assess how each of these effects could give rise to a hazardous scenario and the resultant risks.
3. Where the risks are unacceptable, identify mitigations to reduce the risks to an acceptable level.

A key part of the *context* for such an argument is to identify the scope of *proof of concept* exercise. For example, for an airborne equipment item, to identify to what types of aircraft will the equipment be fitted, and to what proportion of aircraft in a particular region will it be fitted. This *context* is required to determine, for example, the effect on controller workload in the event of a failure of the equipment.

In the aircraft domain, there is already provision for flight testing and endurance testing, which can be considered to be forms of the proof of concept approach. However, it should be noted that the full approach is wider than any individual existing technique.

> The complexity of Flight Management Systems (FMS) makes it infeasible to test them exhaustively prior to introduction into service. Fokker F100 VNAV was improved following service entry using feedback from revenue flights; the improved version was then tested, as a proof of concept exercise, before introduction as a final version. Future FMS system development and introduction would benefit from wider application of this proof of concept approach, supported by logical argument to (a) identify where proof of concept would be most effective and (b) justify safety of execution of the proof of concept. (In this case, the testing could be undertaken in revenue service, because the Fokker FMS 100 is not used as a primary navigation tool; however the proof of concept approach can also be applied without needing to undertake testing in revenue service.)

[21] Where the *proof of concept* is undertaken in a simulated environment, then the first argument is not necessary.

3.4.3 Option 7: Enforce existing rules and improve existing processes

The logical argument approach provides a framework by which the current good practice and established approaches can continue to be applied, while also providing the flexibility to adapt where appropriate to improve the overall approach. The approach also provides techniques which support identification and addressing of "gaps" at the interfaces between domains, which is critical to improving existing processes.

In addition, key individual issues identified so far during the ASCOS project (including the *bottlenecks* and *shortcomings* and other issues raised by the stakeholders) are addressed as described in section 3.5.

3.4.4 Option 8: Cross-domain fertilisation

The logical argument approach provides a framework within which different approaches can be introduced where relevant. The following approaches may be useful when attempting to certify changes to the *TAS*:

- five-part logical argument as presented in section 3.2 – this provides a useful framework for making safety arguments spanning the whole development lifecycle;
- three stage approach to "certification" of a flight operator (see template in Appendix D):
    - certification - determining that the applicant is a competent organisation with appropriate procedures in place;
    - licensing – granting permission to operate specific aircraft over specific routes;
    - oversight – monitoring to ensure that the operation remains safe.

    Existing requirements for certification [meaning here the entire process of bringing operations into service and monitoring the operation] of flight operators are specified in detail and are effective in ensuring safety of flight operations. The existing approach follows the 5 phase ICAO process (although this is only advisory). The logical argument approach allows these requirements to be retained, while also allowing the flexibility of using a different approach where the applicant has specific needs which are not covered in the requirements. It may also be beneficial to adopt this approach when certifying other organisations involved within the TAS.

- the Level of Involvement (LoI) concept, where certain manufacturers holding design organisation approval (DOA) are granted the privilege to approve some major changes where:
    - these are repetitions of similar previously approved changes;
    - the organisation has demonstrated a suitable level of competence; and
    - EASA has agreed that it does not need to be involved.

    This concept is the subject of research within EASA, and the subject of a presentation given to the ASCOS project [29].

- product-based approach as adopted in the rail industry where the *argument* is partitioned into three distinct types of safety case (generic product, generic application and specific application) - see Appendix E.2.3 for more details.

3.5 Addressing existing regulations and processes

Several exercises have been undertaken to identify areas where the existing regulations and processes detract from an optimal *certification* process. This includes:

- the work undertaken by ASCOS to identify *bottlenecks* and *shortcomings* in existing regulations;
- meetings between the ASCOS team and stakeholders – this includes the meetings of the User Group and a separate meeting between ASCOS and EASA in April 2013;
- a questionnaire prepared as part of this study and distributed to ASCOS partners and the User Group.

The findings of these exercises are presented in Appendix C.

Although it is not possible for the proposed certification approach to fully address all these issues, it does provide a flexible framework which helps to address these issues in the following way.

1. It supports innovation[22] through introducing flexibility in *certification*; this is done by allowing the *argument* to be structured according to the needs of the application, including where existing standards are insufficient or where products have been pre-developed to different standards (COTS).
2. It supports early and better communication and integration between *domains* through:
    a. encouraging involvement of stakeholders early in, and throughout, the 10 step process;
    b. modularisation of the *argument* and thorough identification and definition of *assurance contracts* between those *modules*;
    c. introduction of the concept of an *argument architect* to oversee the overall development of the *argument*.
    d. to ensure that safety requirements are not lost at the boundary between domains.
3. The steps defined in point 2 above also support the appropriate involvement of the regulator(s) throughout the process and help to address concerns over poor interface management.
4. It identifies areas where existing supervision or oversight is weak (as evidence will not be available to support that part of the argument) and can be used as a framework to support development of appropriate regulatory models and oversight for novel concepts.
5. It provides a framework to describe the *argument* across the whole system lifecycle, thus ensuring that safety throughout all system lifecycle phases is adequately considered.

Whilst it is considered that the approach offers opportunities for improving efficiency in current processes, it is necessarily geared more towards ensuring that future certification processes are effective and efficient in addressing innovation. Innovation is seen as having a negative impact on reducing shortcomings in the certification process, but these must be overcome if the benefits of innovation are to be realised without jeopardising the safety of air traffic. Innovation can also drive up the scale and complexity of the safety assurance and this in turn puts a greater demand on the availability of expert resource within the community.

[22] Terms underlined in this list correspond to the issues identified in Figure 7.

4 Staged Application of the Approach

This section defines stages for application of the approach. The stages are defined for application to the "real" *TAS*. The first application will be in the ASCOS case studies which form WP4, whereas validation of the approach is addressed in WP5. It is intended to use the experience of the case studies and the validation to refine the approach; this will include refinement of the stages defined here.

For the case studies, selected ASCOS User Group members will be involved to provide assistance and act as "representative" stakeholders and acceptance authorities.

The stages of the approach are listed below. The following sections provide more detail for each stage. The triggers for initiating the approach are identified in section 3.1.2.

1. Define the change
2. Define the certification argument (architecture)
3. Develop and agree certification plan
4. Specification
5. Design
6. Refinement of argument
7. Implementation
8. Transfer into operation – transition safety assessment
9. Define arrangements for continuous safety monitoring
10. Obtain initial operational certification
11. Ongoing monitoring and maintenance of certification

These stages are aligned with the lifecycle stages and the generic argument presented in section 3.2. In reality depending on the nature of the change (e.g. whether or not the change may be considered 'minor' or 'major') some of the stages may be skipped or combined, but the principles remain the same for each stage.

However, it is important to note that the responsible party for each stage of the argument may be different and this means that there can be *assurance contracts* between the stages as well as between the various components of the system or service. For example an *assurance contract* will exist between the manufacturer of an aircraft and the operator / maintainer of the aircraft.

If *progressive certification* is adopted, acceptance would be obtained from the relevant authorities following each of the stages listed, in order to derisk the achievement of operational certification.

Note the following sections refer to safety assessment processes as summarised below and detailed in the related ASCOS work packages, but can also be aligned with the processes for Functional Hazard Assessment (FHA), Preliminary System Safety Assessment (PSSA) and System Safety Assessment (SSA) as described in for example EUROCAE ED-79A / SAE ARP4754 [34] or SESAR Safety Reference Material [37].

A priori safety assessments – WP3

- ASCOS WP3 is developing safety based design methods and tools that enable the handling of current, emerging and future risks. These methods and tools address the TAS, and support the derivation of Safety Objectives and Safety Requirements for any proposed change within the TAS (e.g. new technologies, operations, systems and/or products). The current approach is to use an improved Causal model for Air Transport Safety (CATS) [38, 44] and the Future Aviation Safety Team (FAST) methodology [39, 40, 41] to support the hazard identification and hazard classification processes related to the (functional) specification and (system) design associated with the proposed change. This implies that the WP3 methods and tools initially focus on supporting the stages 4, 5 and 6 of the certification approach, as part of the 'a priori risk assessment' before implementation of the change (i.e. usage to support Cl 1 – Cl 3).

A posteriori safety assessments – WP 2

- ASCOS WP2 is developing a process and tools for multi-stakeholder Continuous Safety Monitoring (CSM), using a baseline risk picture for the TAS (i.e. including all domains and their interactions) [43]. The CSM process supports a posteriori risk assessments by establishing the framework for collecting data. Safety Performance Indicators (SPIs) are being specified to monitor the safety in service [42]. This implies that the WP2 methods and tools initially focus on supporting the stages 8, 9 and 10 of the certification approach, as part of the 'a posteriori risk assessment' (i.e. usage to support the Cl 5).

4.1 Stage 1: Define the change

This stage is focussed on ensuring that the proposed change[23] to the *TAS* is fully understood. This includes defining / identifying:

- the overall goal of the change;
- definition of the change to be made, including the intended functions and an operational concept;
- initial high level architecture for the change, sufficient to allocate requirements between the *domains* of the *TAS*;
- definition of the time frame for the actual implementation of the change (target year);
- what Areas of Change (AoC)[24] within the *TAS* are expected within the defined time frame;
- which of the AoCs, expected within the time frame, would possibly affect the change to be made;
- what part(s) of the system will be changed (including operational processes, products, roles for human actors), or affected[25] by the change – this includes, but is not limited to, identifying the *domains* changed or affected;

[23] The term change is used here to include any change to the overall *TAS*, including introduction of new operations, concepts, processes, systems, or new aircraft or equipment, or making changes to existing elements of the system.
[24] The concept of Area of Change (AoC), which was introduced by the FAST, is defined as any (future) phenomenon that will affect the safety of the aviation system either from within or from important domains external to aviation

- what organisations are involved in making the change (e.g. introduction of a new ATM system will involve, at least, the ANSP and the equipment manufacturer);
- how the *external environment* may be affected by the change;
- initial *argument architecture* related to the change based on the above including identification of *assurance contracts*
- what existing regulations, certification specifications, standards, AMCs or other relevant guidance material are applicable to the change;
- what requirements (including safety requirements) the change needs to fulfil[26].

At the early stages of a programme, there is usually a high degree of uncertainty over some of the details of the change. It is important to document what is known (and what is assumed) about the system from the outset so that subsequent changes can be fully assessed. As the programme develops, and the change is defined in greater levels of detail, corresponding requirements and assumptions must be allocated to the relevant components of the system and any interfaces between these components fully defined.

The other key issue at this stage relates to who is proposing to make a change, for example it could be an ANSP, aircraft manufacturer, maintenance contractor, avionics equipment supplier, etc. Often changes can be confined to the boundary of concern of the party making the change, but this needs to be verified. The initial development of the *argument architecture* is used to identify the interfacing *arguments* and capture any relevant *assurance contracts*. Any change that alters an *assurance contract* will need to involve the owner of the interfacing *module(s)* to address the change impact and if necessary renegotiate the *assurance contract*.

The purpose of identifying regulations, standards, AMCs etc is to guide the structuring of the safety *argument*, and support identification of any *assurance contracts*. For example, if the change includes introduction of a new item of airborne equipment covered by the certification specifications (CSs), this part of the argument may be *compliance-based*, showing compliance with the relevant CS. (Of course, the argument may adopt a performance-based approach, in line with Option 2 – see section 3.4.1.)

All the information identified in this stage becomes input to the definition of the safety argument.

4.2 Stage 2: Define the certification argument (architecture)

This stage is focussed on developing the initial certification *argument* which will be made for the change. The information gathered in Stage 1 should be sufficient to define the top level claim of the *argument* and the necessary context for that claim.

---

[25] I.e. an element which is not actually changed, but which the change may have an impact on – e.g. change of an airborne equipment item may affect the flight crew even if the procedures which they follow may remain unchanged.
[26] It could be argued that safety requirements are part of the regulations, standards, etc which apply, but these have been highlighted separately because they are critical to the argument.

The generic *argument* to be adopted should be chosen and developed into an *argument architecture*. It is proposed that, for each of the case studies, the generic *argument* outlined in section 3.2 is initially adopted, unless it is evident from the outset that an alternative *argument* is appropriate. (In the event that alternative top level *arguments* are identified during the case studies, these will be documented in the presentation of the refined approach.). Note however, that variation in the *argument* approach is not likely to affect the modularisation of the argument as this is driven more by the existing commercial and physical partitions within the *TAS*. It may affect the links between modules but this should be avoided especially if it affects an existing *assurance contract*. At this stage the *argument* should identify any potential impact either on or from existing *assurance contracts* or *modules* outside the initial scope of the change. Note the full impact may not be realised until later (e.g. during implementation) but consideration should still be given to any known impacts at this stage, as they may alter or undermine key *assumptions* in the design of the change.

The development of the *argument architecture* should follow the principles identified in section 2.2 and section 3.3. The architecture will follow existing established *certification* approaches where these remain appropriate (e.g. compliance with CSs for airborne equipment) while ensuring that any consequences of using this approach are fully understood and managed – for example the need to establish that the CS remains applicable within the *context* of the specific change.

The argument should then be developed by the *argument architect* (see section 2.2.1). It remains the *argument architect's* responsibility to maintain the *argument* throughout the lifetime of the change.

The level to which the argument can be developed at this stage is limited until the assessment activities associated with Specification (Stage 4 – section 4.4) and Design (Stage 5 – section 4.5) have been completed. However, it is important to develop the initial argument to provide a basis for development and agreement of the certification plan. The argument is then refined (Stage 6) – see section 0.

4.3 Stage 3: Develop and agree certification plan

The role of the certification plan is to show how the certification *argument architecture* will be developed and substantiated with evidence to the point where it can be presented for acceptance by the relevant authorities.

The certification plan presents the *argument architecture*, along with the certification activities to be undertaken, including how impacts, if any, on existing *assurance contracts* will be addressed.

It is recognised that a given change may require endorsement from multiple authorities, each of which may only be competent to endorse the residual risk for part of the system. Thus it may not be possible for any one authority to endorse the top level of the *argument*. Consequently it is necessary for the certification plan to clearly define the parts of the *argument* which require endorsement by each authority.

The certification plan is presented to the relevant authorities and other stakeholders, to gain their agreement that, if the plan is followed and the evidence is presented, they will accept the change into service. Although lack of agreement at this stage does not prevent progress to later stages, the benefit of gaining agreement is to reduce the risk to the certification programme at later stages. This approach can be developed further into

progressive certification where agreement is obtained for the argument progressively as the individual claims (Cl 1 through to Cl 5 in the generic argument in section 3.2) as they are completed.

Stakeholders / authorities all have different perspectives and often introduce differing / additional requirements. These requirements may all (or mostly) be beneficial, but they introduce significant cost increases if they are introduced progressively through the project.

4.4 Stage 4: Specification

This stage is focused on demonstrating that Cl 1 of the generic argument is met, namely that the change is specified to achieve an acceptable level of safety. As described in section 3.2, this focuses on the behaviour of the changed system in the absence of failure – i.e. does the changed system sufficiently mitigate the pre-existent hazards within the *TAS*? As part of this stage the argument for Cl 1 is fully developed and substantiated with relevant evidence.

Safety assessment in this stage is used to identify the pre-existing hazards relevant to the system[27] and assesses the consequences of these hazards on the safety of the *TAS*. This assessment is used to derive definitions of:

- the safety objectives for the system;
- the safety requirements which specify what the system is required to do (not how it does it) in order to achieve the safety objectives;
- the degree of assurance required that the system will meet its requirements;
- any additional functionality requirements or assumptions to capture any external means of mitigating the consequences of the hazards caused by failure of the system.

The techniques developed as part of WP3 provide support to this assessment. However, it should be noted that many of these techniques focus on the assessment of hazards resulting from system failure.

At this stage the modularisation is reassessed to make sure all relevant external modules and any assurance contracts are linked and impacts identified. This will include an initial assessment of claim / context matching based on the context captured to support Cl 1, especially any scoping statements, assumptions and dependencies or other claims that are needed to support the argument under Cl 1.

The safety assessment in this stage broadly aligns with the FHA process as further described in the documents referred to in section 4.

[27] At this stage the assessment is performed independent of the implementation of the system and so failures of the system relate to specification-level functions only. However, it should be noted in some cases consideration must also be given to concept of operations, modes of operation or other operational definitions which depict the intended use of the system.

4.5 Stage 5: Design

This stage is focused on demonstrating that Cl 2 of the generic safety argument is met, namely that the logical design for the change satisfies the specification derived within Cl 1. As part of this stage the argument for Cl 2 is fully developed and substantiated with relevant evidence.

Safety assessment at this stage considers what the elements of the logical design need to do to ensure safety and the degree of assurance required. Requirements derived during this stage are set without necessarily prejudging how that design should be physically implemented. However, the assessment also needs to consider the achievability of any requirements and therefore must consider whether the requirements can be met (at least in principle) by the preliminary design.

This stage identifies hazards resulting from failures of the system and produces a set of Design Safety Requirements[28] (DSRs) which define what each element of the design has to do, in terms of functionality and performance, in order to mitigate these hazards. This stage also demonstrates that the design would actually work as intended under all expected normal and abnormal conditions. The assessment should also identify high level causes of system-generated hazards and specify Safety Assurance Requirements for each element of the design.

The main output of the safety assessment is as follows:

- Design Safety Requirements for each element of the logical architecture, as necessary to provide the functionality and performance specified in the specification stage
- Safety Assurance Requirements for each element of the logical architecture, as necessary to satisfy the level of assurance specified in the specification stage
- additional Design Safety Requirements (or assumptions, where appropriate) to capture any internal means of mitigating the causes of the hazards arising from failure of the system.

The safety assessment in this phase broadly aligns with the PSSA process as further described in the documents referred to in section 4.

4.6 Stage 6: Refinement of Argument

Following the detailed assessment undertaken in stage 4 (Specification) and stage 5 (Design), the detail of the safety *argument* is updated to correspond to the safety requirements derived and the more detailed understanding of the system architecture which has been developed.

This update includes ensuring that all relevant external *modules* and any *assurance contracts* are linked and impacts identified. This will include reassessment of *claim* / *context* matching based on the *context* captured

[28] Both the SESAR Safety Reference Material and ED-79A / ARP4754 use broad definitions of the term safety requirement. Safety is assured by both design requirements AND assurance requirements being satisfied. Not all design requirements are safety related, but it is essential to the safety argument that the safety related design requirement are both correctly and completely defined, and subsequently proved in the implementation.

to support Cl 2, especially any further scoping statements, *assumptions* and *dependencies*, including those identified as part of following WP3 methods, or other *claims* that are needed to support the *argument* under Cl 2.

At this stage it may be necessary to update the certification plan to correspond to the updated argument. In this event the plan should then be resubmitted to the relevant authorities to confirm their continued acceptance of the approach.

For simplicity this stage is shown as a single event at this point in the process. In practice the refinement of the argument is ongoing through the certification programme.

4.7 Stage 7: Implementation

This stage is focused on demonstrating that Cl 3 of the generic safety argument is met, namely that the physical implementation of the logical design (defined in Cl 2) for the change is complete and correct. As part of this stage, the argument for Cl 3 is fully developed and substantiated with relevant evidence, and captured in the module or modules defined to support the system implementation.

The principle aim of safety assessment at this stage is to demonstrate by a combination of analysis and testing, that the (as-built) system[29] meets the safety requirements. Depending on the complexity of the design it may be necessary to further derive a detailed set of safety requirements for the physical system design; these are obtained by allocating the Design Safety Requirements for the logical design (derived in the design stage, as above) on to the physical architecture.

This stage also derives detailed Safety Assurance Requirements for the physical architecture and shows that these are met. It is at this stage that the system implementer often encounters a major problem – i.e. the limited ability of, inter alia, test-based validation & verification to show, with sufficient confidence, that the required safety integrity properties of the system have actually been satisfied in practice. An assurance based approach is often followed to provide this demonstration. (One such approach is defined in the UK CAA SRG CAP670 [24] and the associated AMC [33] for the SW01 requirement.)

At this stage the modularisation is updated significantly to address the physical architecture (and commercial boundaries) of the implementation. This may result in additional *modules* being defined and added to the modular architecture to manage the complexity and scale of the expansion of Cl 3 down to the components of the system. This in turn may identify further external *modules* and *assurance contracts* that are linked to the internal modules. As previously any impact on existing *assurance contracts* will need to be identified and addressed. This will include assessment of *claim* / *context* matching based on the *context* captured to support Cl 3, especially any further scoping statements, assumptions and dependencies, including those identified as part of following WP3 methods, or other *claims* that are needed to support the *argument* under Cl 3.

[29] Still recognising that the system includes people, processes and equipment.

The safety assessment in this phase broadly aligns with the SSA process as further described in the documents referred to in section 4.

4.8 Stage 8: Transfer into operation assessment

This stage is focused on demonstrating that Cl 4 of the generic safety *argument* is met, namely that the transition to introduce the change is acceptably safe. The two main elements of this are to confirm whether:

a. the fully proven change is ready to be brought into operational use
b. that the introduction of the change can be achieved without affecting the overall safety of the system while the change is being introduced.

The following aspects need to be considered.

- Preparation for operation, including publication of operational and engineering procedures, provision of resources (people, equipment spares, maintenance facilities etc) and training of operational and technical personnel.
- Implementation of arrangements for safety management, change management, configuration control etc – for changes to an existing operation, these may already defined within the relevant organisation(s)'s management systems.
- Confirmation that the process of switching over from the old systems to the new systems has been fully planned and resourced. This should include switchover procedures, allocation of responsibilities and the training / briefing of all personnel involved.
- Assessment and mitigation of all hazards associated with switch-over from the old systems to the new systems. This assessment should include safety assessment of the switchover and should result in the additional procedures, allocation of responsibilities and training / briefing of personnel necessary to prevent (as far as possible) things going wrong, and to take the appropriate action should something go wrong.

As part of this stage the *argument* for Cl 4 is fully developed and substantiated with relevant evidence. The scale and complexity of the transition *argument* can very hugely dependent on the nature of the change, but is usually more comprehensive for novel, complex or large scale changes, and especially those that require coordination of multiple parties. This stage is particularly relevant for the *proof of concept* option as this could be a key component of a transition *argument* for new technology for example.

For simple arguments the modularisation may be unaffected by the transition; however, it is more likely that the *argument* will open new links with extant modules in order to facilitate the transition. If the transition is complex then this may result in additional modules being defined and added to the *argument architecture* to manage the complexity and scale of the expansion of Cl 4. This in turn may identify further external *modules* and *assurance contracts* that are linked to the internal modules. As previously any impact on extant *assurance contracts* will need to be identified and addressed. This will include assessment of *claim* / *context* matching based on the *context* captured to support Cl 4, especially any further scoping statements, *assumptions* and

dependencies, including any requirement for additional temporary mitigations during transition, or other *claims* that are needed to support the *argument* under Cl 4.

4.9 Stage 9: Define arrangements for continuous safety monitoring

This stage is focussed on demonstrating that Cl 5 of the generic safety *argument* is met, namely that arrangements are in place to ensure that the change is demonstrated to be acceptably safe in operational service.

As part of this stage, the *argument* for Cl 5 is fully developed and substantiated with relevant *evidence*, and captured in the *module* or *modules* defined to support the ongoing operations. The argument here will include and substantiate the application of WP 2 methods [43]. However the argument for this stage necessarily takes a different form from the argument for the previous stages, because it is about demonstrating that processes are in place, rather than demonstrating that evidence has been collected.

At this stage it is necessary to show that:

- continuous safety monitoring (CSM) collects the appropriate metrics to confirm the results of the safety assessments undertaken to support the earlier stages of the argument;
- processes are in place to report and investigate all safety-related incidents and to ensure that appropriate corrective action is taken;
- processes are in place to carry out safety assessment of any interventions (e.g. maintenance) to ensure that the associated risks are known and acceptable.

4.10 Stage 10: Obtain initial operational certification

At this stage, the evidence generated in earlier stages is presented by the applicant to the relevant authorities in order to obtain permission to introduce the change into service.

Assuming that

a. the authorities have previously accepted the certification plan (see stages 3 and 6) and
b. the later stages (stages 4, 5, 7 – 9) have undertaken the activities defined in the plan and produced evidence to support the certification argument

then the authorities need to undertake sufficient review of the evidence to confirm that the change is acceptably safe.

The approach does not replace existing criteria, and these would still be used by the authorities to review the evidence where these criteria apply. However, where existing criteria do not apply, the argument would identify the criteria to be applied in assessment of the evidence, and these would have been agreed with the authorities during development of the certification plan.

If the approach has deviated from the certification plan, then the applicant should consider repeating stage 3 in order to confirm that the revised approach will be acceptable.

4.11 Stage 11: Ongoing monitoring and maintenance of certification

Following introduction into service, the monitoring arrangements defined in stage 9 must be implemented. The certification *argument* must be updated at regular intervals to confirm that the changed system continues to achieve the relevant requirements. The intervals for update and recertification should be specified by the certifying authority.

Where further changes are made to the *TAS* it is important that the requirements of all previous changes are taken into account, so that safety mitigations are not lost in subsequent changes. This is where it becomes particularly important to identify an *argument architect* for the certification *argument*, to prevent issues being lost at the boundaries between *domains* and responsibilities.

5 Conclusions

The objective of the ASCOS project is to develop novel certification process adaptations and supporting safety driven design methods and tools to ease the certification of safety enhancement systems and operations.

This study focuses on the certification process. The aim is to take certification beyond the current state of the art using an integrated approach which considers the *total aviation system* across the whole lifecycle of development, deployment and operation and which is applicable to the innovations envisaged within future developments of aviation. The consideration of the whole system includes people and processes as well as the equipment; it also addresses the potential for gaps at the interfaces between different elements of the system.

Previous work has laid the foundations for the proposed certification approach [1]. Current regulations and practices in the various domains of the *total aviation system* (*TAS*) were reviewed to identify potential *bottlenecks* and *shortcomings* in the efficacy of the current processes. Eight possible options for improvement were defined and evaluated, of which four were considered to have the most promise:

- Option 2: Change between performance-based and compliance-based or vice versa
- Option 6: Proof of concept approach
- Option 7: Enforce existing rules and improve existing processes
- Option 8: Cross-domain fertilisation

This report presents the proposed certification approach and the steps for application of this approach to the case studies. The approach is to build a logical *argument* for the *certification* of any change to the *total aviation system*, supporting the top level *claim* that the change is acceptably safe. The *argument* captures the definition of the change including all relevant *context* (including acceptance criteria and assumptions). The *argument* is decomposed into supporting *claims* until the *claims* can be directly supported. The level of decomposition is limited initially to that necessary to support definition of the interface between the TAS domains, and to dovetail with the existing domain certification approaches. This framework advances the state of the art by driving unification of the *argument* across all *domains*.

The approach also includes the concept of modularisation, where the overall *argument* is decomposed into manageable *modules*, each of which encapsulates the *argument* for a particular component of the overall *argument*. The boundary of each *module* represents the public view of the *module* and includes a definition of the *claims* made in the *module* and associated *context*, *caveats* and *dependencies*. The *module* boundary definition provides all the information necessary to facilitate linking with other *modules*. Definition of an interface then makes it possible to establish *assurance contracts* between *modules*; this approach is particularly useful when *modules* are being developed by different organisations as it allows a clear definition of the responsibilities of each. The overall *argument architecture* consists of the *modules* and the relationships between them, including the *assurance contracts* defined.

"Gaps" between elements of the *TAS* are one key area of concern, considered to be insufficiently addressed in the current state of the art. Careful modularisation and definition of *assurance contracts* is key to advancing

the state of the art in this area. Modularisation drives identification of these interface issues and definition of *assurance contracts* establishes responsibility for ensuring that these issues are correctly managed both during development and throughout the lifetime of the system.

Even with effective modularisation, *arguments* can become very complex and include significant elements which are outside the responsibility of the applicant proposing the change. To reduce this complexity the proposed approach will avoid unnecessary development of detailed arguments where existing certification practices are sufficient. Nonetheless effective application of the approach requires an *argument architect* to take the overall responsibility for the development and maintenance of the *argument architecture* across all the affected *domains*. (It is recognised that the *argument architect* is not necessarily responsible for endorsing the overall argument; in fact there may be no single authority competent to do this; careful planning is therefore needed to confirm how endorsement is achieved.) The responsibility of the *argument architect* extends beyond the introduction of the change, as key elements of the *argument* will require confirmation throughout the lifetime of the system. There are a number of options for who would take the role of *argument architect* and, indeed, the role may transfer between parties during the lifecycle of the change. Where the change primarily affects a single domain, the applicant may be best placed to act as *argument architect*, but where the change is more widespread someone with wider responsibility would be needed to ensure that the implications of the argument are followed through in all domains. This needs to be explored further.

The logical *argument* approach is flexible, in that it allows retention of existing *certification* processes within individual *domains* (thus implementing Option 7), while also ensuring that the *context* in which the existing certification is developed is fully considered within the overall *argument*. The flexibility also allows for alternative approaches to be taken where the change being introduced is not covered by existing specifications, thus supporting innovation in process or technology, as required by the overall aims of the ASCOS project. This may involve changes between *performance-based* and *compliance-based* approaches (Option 2); it may also introduce approaches from other aviation domains or from other industries (Option 8). It also provides the flexibility to introduce the *proof of concept* approach (Option 6).

The proposed approach has taken into account industry concerns including the identified *bottlenecks* and *shortcomings* and the directly expressed concerns of the ASCOS User Group. Although it is not possible for the approach to directly resolve all the issues identified, it does provide a framework which supports them being addressed. In particular, the approach:

- provides flexibility to support innovation in (a) technologies and concepts and (b) certification approaches;
- provides a framework to improve communication and integration between domains;
- provides a process to support engagement of stakeholders (including authorities) throughout the lifecycle;
- builds on the approach taken in the SESAR Safety Reference Material.

The trigger for application of the approach may come from a business need, a change to the environment or from continuous safety monitoring. However, the approach is not applied where activities are covered by the existing management system.

The logical *argument* approach is well established in the ATM *domain* and in other industries and shows the most promise to ease the certification of safety enhancement systems and operations; however it should be noted that it is intended to retain existing approaches where they remain appropriate and to use the argument framework to integrate the approaches taken in each domain. (This approach is often supported with the use of a graphical notation.) However, as there is no common or widely agreed methodology for constructing such arguments and there is no clear practice on interfacing arguments between domains or between lifecycle phases, the application proposed here seeks to introduce a number of innovations, including:

- provide a basis for unification of multiple *certification* approaches from multiple *domains* within a single logical *argument*;
- encourage wider application of logical *arguments* to address novel systems and concepts,
- develop a more robust approach to safety argument construction;
- utilise safety arguments to support the cross fertilisation of certification approaches;
- apply the principles behind safety argument modularisation to manage interfaces between different parts of the system and lifecycle phases.

The approach is supported by generic *argument* templates from which detailed *arguments* can be developed. These template *arguments* are structured to address the whole development lifecycle, from development of a specification, through to monitoring of the system throughout its operation. One of the templates is a high level argument which has been successfully used in ATM applications and which forms part of the guidance material published by EUROCONTROL and SESAR.

Additional templates have been developed to capture arguments for:

- certification, licensing and oversight of flight operations by an operator;
- application of the proof of concept approach;
- a compliance-based approach where the context of the relevant standards is explicitly assessed.

Another area of support for the framework is in the development of a consistent language to express the concepts involved. Already within the development of the approach, inconsistent terminology has been shown to be a significant barrier. Building on the work of the OPENCOSS project, this study has developed a lexicon of terms to describe the concepts involved and this lexicon will be further developed in subsequent activities.

The proposed approach will be applied to the ASCOS case studies (WP4) and validated (WP5) through comparison with the approach taken in previous certifications within the aviation industry. This experience will then be used to further refine the approach and develop further guidance and template arguments to allow application of the approach across the *total aviation system*.

The steps for application of the approach identify the activities to be undertaken throughout the lifecycle of the change, from the initial definition of the change, through the planning and execution of the certification approach, including engagement with the relevant authorities. The steps also consider the transition when the proposed change is put into operation and the continuous monitoring of the changed system to ensure that the claimed level of safety is indeed met. The guidance includes the interaction with the ASCOS work packages:

- WP3 (safety risk management) which provides the safety assessment methodology framework to support the (*a priori*) risk assessments required to support the overall argument;
- WP2 (continuous safety monitoring) which provides the monitoring framework to support the (*a posteriori*) risk assessment of the change in operation.

Key issues which need to be considered in the application of the approach include the following.

- Carefully define the top level of the argument to be used and the context in which it is expressed, including the context (including system definition and acceptance criteria) as these drive the rest of the argument.
- Identify who is best placed to act as the *argument architect*, who is custodian and maintainer of the arguments, both during development of the change and through the lifetime of the changed system.
- Establish a consistent terminology so that concepts are understood in the same way by all parties involved.

In conclusion, this document proposes an outline certification approach which takes aviation certification beyond the current state of the art. The approach is applicable to any change within the *total aviation system*. This approach considers the whole lifecycle and is flexible enough to accommodate and integrate the existing certification approaches taken in the individual *domains* of the system and to allow adaption of existing approaches, or introduction of new approaches, where required by innovative changes, or elements of changes.

The proposed approach is “modular” and alleviates identified *shortcomings* and *bottlenecks*, such as the narrow scope of safety cases for individual domains and gaps at the interfaces between different domains within the total aviation system. The approach builds on a proactive risk assessment methodology that can be applied to emerging and future risks, and shows how an argument can be constructed to address these risks in order to adequately and safely specify, design, implement and monitor a proposed change in sufficient detail.

The proposed approach will be applied within ASCOS case studies and validated, and the experience gained will be used to further improve the proposed approach for application across the *TAS*. Training and guidance material will be initiated (as part of another work package) to ensure accessibility of the certification approach outside the ASCOS team.

References

#	Authors(s), Title, Year
1	ASCOS: Current certification processes and practice in aviation: towards an assessment framework, 2013
2	ACARE; European Aeronautics Vision for 2020: Meeting society's needs and winning global leadership, Report of the Group of Personalities, ISBN 92-894-0559-7, 2001.
3	OPENCOSS D2.2 Version 2.3: High-Level Requirements on the OPENCOSS Platform, 2012
4	OPENCOSS D4.1 Version 1.0: Baseline for the Common Certification Language, 2012
5	OPENCOSS D4.2 Version 1.0: Detailed Requirements for the Common Certification Language, 2012
6	OPENCOSS D5.1 Version 1.0: Baseline for the Compositional Certification Approach, 2012
7	EN50129:2003: Railway applications – Communication, signalling and processing systems – Safety related electronic systems for signalling
8	ISO 26262-2:2011: Road vehicles – Functional Safety – Part 2: Management of Functional Safety
9	EUROCONTROL: E-OCVM3: 2010: European Operational Concept Validation Methodology Version 3.0 Volume 1.
10	ASCOS WP1.3 Partner Questionnaire Version 1.3, 2013
11	ASCOS D3.1 Version 1.0: Total aviation system safety assessment methodology, 2013
12	ASCOS D2.1 Version 1.3: Framework safety performance indicators, 2013
13	ASCOS D2.2 Version 1.3: Total aviation system baseline risk picture, 2013
14	Origin Consulting, GSN Community Standard, Version 1, 2011
15	EUROCONTROL DAP/SSH/091 Edition 2.2: Safety Case Development Manual, 2006
16	Dr TP Kelly COMSA/2001/1/1: Concepts and Principles of Compositional Safety Case Construction, 2001
17	J. Monso, B. Rabiller: SESAR Proof of Concept supporting document, SESAR P16.01.04, deliverable D4
18	J. Monso, B. Rabiller: SESAR Guidance Material to execute Proof of Concept, SESAR P16.01.04, deliverable D6
19	S. Wagner et al: A Case Study on Safety Cases in the Automotive Domain: Modules, Patterns and Models, 2010
20	S. Bates et al: Safety case architectures to complement a contract-based approach to designing safe systems, 2003
21	J. Fenn et al: Safety Case Composition Using Contracts – Refinements based on Feedback from an Industrial Case Study, 2007
22	Network Rail: Product Acceptance (http://www.networkrail.co.uk/aspx/3262.aspx), accessed 23/10/2013
23	UK CAA Safety Regulation Group CAP760 First Edition, Amendment 2010/01: Guidance on the Conduct of Hazard Identification, Risk Assessment and the Production of Safety Cases, 2010

#	Authors(s), Title, Year
24	UK CAA Safety Regulation Group CAP670 Third Issue, Amendment 1/2013: Air Traffic Services Safety Requirements, 2013
25	Ebeni, Summary Report – Safety Assurance of the Draft Specifications for the Use of Military UAVs as OAT Outside Segregated Airspace, 2006
26	EUROCONTROL RVSM 691 Version 2.0: The EUR RVSM Pre-Implementation Safety Case, 2001
27	I. Habli, T. Kelly: Achieving Integrated Process and Product Safety Arguments *in* The Safety of Systems: Proceedings of the Fifteenth Safety-critical Systems Symposium (Springer: 2007)
28	PD CLC/TR 50506-1:2007: Railway Applications – Communication, signalling and processing systems – Application Guide for EN50129 – Part 1: Cross-acceptance
29	F. Copigneaux (EASA): Level of involvement in product certification, 2012 (presented to ASCOS 19th April 2013)
30	EASA CS-25 Amendment 3: Certification Specifications for Large Aeroplanes, 2007
31	EUROCONTROL Edition 1.0: Safety Assessment Made Easier – Part 1 Safety Principles and an Introduction to Safety Assessment, 2010
32	EUROCONTROL SAF.ET1.ST03.1000-MAN-01 Edition 2.1: Safety Assessment Methodology, 2006
33	UK CAA Safety Regulation Group: Acceptable Means of Compliance to CAP 670 SW 01 – Guidance for Producing SW 01 Safety Arguments for COTS Equipment, Issue 3, 2010
34	EUROCAE ED-79A: Guidelines for Development of Civil Aircraft and Systems, 2011
35	ASCOS Description of Work – project number 314299
36	William S. Greenwell et al, A Taxonomy of Fallacies in System Safety Arguments, 2006
37	SESAR: Safety Reference Material, Edition 00.02.01, Project ID 16.06.01, 30th Jan 2012
38	B. Ale, L.J. Bellamy, R. Cooke, M. Duyvis, D. Kurowicka, P.H. Lin, O. Morales, A. Roelen, J. Spouge; Causal Model for Air Transport Safety: Final report. Directorate General of Civil Aviation and Maritime Affairs, Ministry of Transport, Public Works and Water Management, 2008.
39	FAST; The FAST approach to discovering aviation futures and associated hazards, Methodology Handbook, Future Aviation Safety Team, 2012.
40	FAST; Areas of Change Catalogue: Ongoing and future phenomena and hazards affecting aviation, compiled by the Future Aviation Safety Team, February 19, 2013.
41	M. Masson, Y. Morier (EASA) and FAST; Methodology to Assess Future Risks - Action EME 1.1 of the European Aviation Safety Plan (EASp, Presented to ECAST 4-12, 11-12-2012
42	ASCOS: Safety performance indicators for the system of organizations in aviation, 2013
43	ASCOS: Improving safety performance in the total aviation system, 2013
44	ASCOS: Risk models and accident scenarios in the total aviation system, 2013
45	European Commission; Flightpath 2050: Europe's Vision for Aviation, Report of the High Level Group on Aviation Research, ISBN 978-92-79-19724-6, 2011.

#	Authors(s), Title, Year
46	ASCOS: ASCOS EASA Workshop Minutes of Meeting (19th April 2013) Issue 1.0, 2013
47	ASCOS: User Group Workshop 1 Minutes (30th October 2012) Issue 1.02, 2012
48	ASCOS: User Group Workshop 2 Minutes (20th September 2013) Issue 1.0, 2013
49	D. Fowler: Getting to the Point: A Safety Assessment of Arrival Operations in Terminal Airspace, Air Traffic Control Quarterly Volume 20 Number 2, 2012
50	S. Thomas, D. Fowler: (Presentation on) Safety Case for the Airborne Collision Avoidance System, Assuring the Safety of Systems – Proceedings of the Twenty-first Safety Critical Systems Symposium, ISBN 978-14-81-01864-7, 2013
51	S. Toulmin, R. Rieke and A. Janik, "An Introduction to Reasoning", Macmillan Publishing, New York, 1979
52	D.N. Walton, "Reasoned Use of Expertise in Argumentation", Argumentation Vol. 3, pp.59-73, 1989.
53	D.N. Walton, "Argumentation and Theory of Evidence", in New Trends in Criminal Investigation and Evidence, vol. 2, pp.711-732, 2000
54	Hahn and Oaksford, A Bayesian approach to informal argument fallacies, Synthese (2006) 152 pp 207 – 236
55	European Commission: 550/2004: Regulation on the provision of air navigation services in the single European sky (the service provision Regulation), 2004
56	AEA Technology (S. Kinnersly) AEAT LD76008/2 Issue 1: Whole Airspace ATM System Safety Case – Preliminary Study, 2001
57	European Commission: 748/2012: Regulation laying down implementing rules for the airworthiness and environmental certification of aircraft and related products, parts and appliances, as well as for the certification of design and production organisations, 2012
58	EASA, ETSO Authorisations (http://www.easa.europa.eu/certification/ETSO-authorisations.php), accessed 13/12/2013.

Appendix A Glossary of terms

Terms presented in this document in *italic* type are defined in Table 2. These meanings have been defined to provide a uniform understanding of underlying concepts across the *total aviation system*.

Terms used in this report are as defined in the relevant ICAO Annexes, EASA CSs or Community regulations such as 549/2004 and 2096/2009. In addition the following definitions, presented in *italic type* in the report, are derived specifically to support the proposed certification approach. These are drawn from the safety argument and modular certification research material referenced in section 2.

Term	*Definition*	*Related terms*
acceptance	*Acceptance* is formal confirmation by the relevant competent authority that the accepted product, system or process may be brought into use. Acceptance may be subject to certain conditions being met; it may also be restricted to a particular application or a defined scope or location of use.	certification
acceptance criteria	*Acceptance criteria* are the requirements which are set by a safety authority and which must be fulfilled before the associated product, process or system will be accepted for use.	
argument	An *argument* is a body of information presented with the intention to substantiate one or more claims through the presentation of related supporting claims, evidence and contextual information.	argument architecture
argument architect	The argument architect is responsible for ensuring that the overall argument is correct and complete and that the argument is maintained throughout the lifetime of the system.	argument, argument architecture
argument architecture	The *argument architecture* is the modular decomposition of the argument into manageable sections.	argument, argument architect
assumption	An *assumption* is a piece of information on which the argument depends but over which the system has no control. For example, an argument may assume that reduced visibility conditions only affect a given airport for 5% of its operational time over the course of a year.	condition, dependency
assurance contract	An assurance contract is a documented formal arrangement between two or more modules within an argument architecture.	
backing evidence	*Backing evidence* shows that processes, tools, techniques and human resources used to produce direct evidence were appropriate, adequate and properly deployed.	evidence, direct evidence

Term	*Definition*	*Related terms*
bottleneck	A *bottleneck* is where existing regulations, although adequate on paper, are not adequately implemented throughout Europe; this may include situations where implementation is not uniform in all States.	shortcoming
certification	*Certification* is interchangeable with acceptance.	acceptance
claim	A *claim* is a logical premise, statement of a requirement or target to be met, usually relating to the system or service under consideration; the intention of an *argument* is to demonstrate that a *claim* is true. In the context of safety *arguments*, the *claim* usually relates to the safety of the system or service.	goal
compliance based	A *compliance based* approach defines detailed criteria to be fulfilled in order to comply with (potentially unwritten or unspecified) performance requirements. This approach has the advantage of making it easier to determine whether the solution is compliant, while potentially constraining implementation.	performance based
compliance management	*Compliance management* is the ongoing process of managing the compliance of a system, process or product with its requirements after it has been accepted.	
condition	A *condition* is something which must become true before an argument is complete. For example, where a safety case is made for a product, it may include conditions about how the product is to be applied / integrated into the overall system.	dependency, assumption, constraint, limitation
constraint	A *constraint* is a *condition* on the implementation of a product, system or service, which is required to maintain the validity of a *claim*. (Note the distinction between a *constraint* and a *limitation*.)	condition, limitation
context	*Context* is additional information necessary to understand a goal or other element of an argument. *Context* can include system scope, safety assurance standards used, environmental limits, etc.	
contradiction	A *contradiction* is where multiple regulations or standards set requirements which are mutually exclusive.	inconsistency, overlap

Term	*Definition*	*Related terms*
dependency	A *dependency* is a piece of information on which an argument depends but which is under the control of another part of the argument or system. For example, an argument about the safety of an ATM system may be dependent upon correct functioning of airborne equipment which is outside the scope of the ATM system.	assumption, condition
direct evidence	*Direct evidence* is based on observable properties of a product (In this context a product is any artefact produced during the lifecycle: thus it includes elements of the operational system, but it also includes specifications, designs etc.)	evidence, backing evidence
domain	A *domain* is one of the parts into which the total aviation system has been divided, shown as an object in the system diagram. The term *domain* is used for the first level of hierarchy within the total aviation system; the term subdomain is used for elements which make up domains.	
evidence	*Evidence* is information or objective artefacts offered in support of one or more claims.	backing evidence, direct evidence
explicit argument	An *explicit argument* is one which is written down in the form of logical reasoning: this may be either in textual or graphical form.	argument, implicit argument
external environment	The *external environment* consists of everything outside the total aviation system which has the potential to affect it.	
goal	The term *goal* is sometimes used interchangeably with the term *claim*.	claim
implicit argument	An *implicit argument* is one where the structure of the argument is assumed. An example of this is in compliance-based certification where it is implied that an item of equipment will be acceptable if it conforms to the specification.	argument, explicit argument
inconsistency	An *inconsistency* is where multiple regulations or standards cover the same subject matter in different ways or require different approaches.	contradiction, overlap
incorrect argument	An *incorrect argument* reaches the wrong conclusion(s) about the claims being made – e.g. it may conclude that the system has a hazardous failure rate better than the target when a significant source of failure has been missed.	argument, invalid argument

Term	*Definition*	*Related terms*
invalid argument	An *invalid argument* has made logical errors in the argument being made – it may reach the right conclusions but for the wrong reasons. E.g. it may conclude from the existence of a test plan that a system meets its requirements, whereas a test plan is insufficient evidence on its own.	argument, incorrect argument
justification	A *justification* is a statement which explains why the argument has a particular structure or has taken a particular approach.	assumption, context
limitation	A *limitation* is a *condition* on the operational use of a product, system or procedure, which is required to maintain the validity of a *claim*. (Note the distinction between a *constraint* and a *limitation*.)	condition, constraint
module	A *module* is a part of an *argument* which has a defined interface, such that changes to the body of the module do not affect the rest of the argument as long as the interface is maintained.	
overlap	An *overlap* is where multiple regulations or standards specify the same requirement.	contradiction, inconsistency
performance based	A *performance based* approach defines the requirements which a product or process must achieve in terms of its impact on the system into which it is introduced, giving the provider flexibility to decide *how* to achieve the requirements.	compliance based
proof of concept	A *proof of concept* is a demonstration whose purpose it is to verify that certain concepts or theories have the potential for real-world application and will be certifiable. For this purpose a *proof of concept* uses a prototype (equipment or procedure) that is designed to determine this potential by testing. This prototype may be an innovative, scaled-down version of the system or operation intended to be developed.	
safety culture	*Safety culture* is the product of individual and group values, attitudes, perceptions and patterns of behaviour which determine the commitment to, and the style and proficiency of, an organisation's health and safety management.	
shortcoming	A *shortcoming* is where existing regulations are either inadequate or simply do not provide the necessary control.	bottleneck
strategy	A *strategy* explains how a parent claim is achieved by the supporting subclaims.	

Term	*Definition*	*Related terms*
total aviation system	The *total aviation system* (TAS) comprises all the organisations, processes, personnel, infrastructure and equipment involved in safely operating air transportation – including aircraft (and all onboard equipment), air operators (including flight crew and maintenance staff), ATM / ANS equipment, ANSPs, aerodromes, air space planning. (See Appendix B.)	

Table 2: Definitions of terms

Appendix B Total System Approach

The *Total Aviation System* (*TAS*) approach is based on the fact that the aviation system components – products, operators, crews, and aerodromes, ATM, ANS, on the ground or in the air - are part of a single network. By considering the whole system, the approach seeks to eliminate the risk of safety gaps or overlaps, and seeks to avoid conflicting requirements and confused responsibilities; it also seeks to streamline certification processes and reduce the regulatory burden.

The term system is used here to mean the whole system, i.e. concepts, equipment, people and processes - not just the physical boxes.

The system can be defined at a number of levels, including:

a. functional specification, including high level functions, operational behaviour and modes of operation;
b. logical design: a high-level architectural representation of the system, independent from the physical implementation. As such it considers the functions provided by the system elements (i.e. human roles and tasks and machine-based functions), but not the equipment, personnel or procedures which provide these functions.
c. physical implementation: the details of equipment (hardware, software and data), people (flight crew, controllers and maintainers), operation and maintenance procedures, training and sectorisation.

As described in the body of this study, different parts of the analysis consider the different levels of the system.

The *TAS* also comprises the following *domains*:

- ATM / ANS equipment: this is the equipment used by the ANSP to provide the air navigation service.
- ANSP: the air navigation service provider is responsible for the provision of navigation information to aircraft with the aim of ensuring safe separation (both between aircraft and between aircraft and terrain); this includes navigation systems, MET systems, AIS, surface movement monitoring – also operation and maintenance of these systems, including training and licensing of controllers and engineers.
- Aircraft manufacture and certification: this covers the certification of the aircraft, including the onboard equipment; this includes design, manufacture, upgrade and instructions for ongoing maintenance, although the actual maintenance is undertaken by the aircraft operator.
- Aircraft operator: this covers flight operations, flight crew selection, training and licensing (including ensuring ongoing competence) and aircraft maintenance in accordance with the procedures laid down by the manufacturer (including selection, training and licensing of maintainers).
- Aerodrome: this covers all aspects of the aerodrome relevant to the TAS (except where already covered by other domains such as ATM / ANS or aircraft / airworthiness) and includes: physical structure (e.g. the runways and taxiways), airfield lighting, security arrangements, management of ground movements – also operation and maintenance of these systems.

- Airspace planning: this covers the strategic planning of the airspace structure and the procedures and protocols for providing air transportation within that airspace structure.

The interaction between these *domains*, and with the *external environment*, is illustrated in Figure 4.

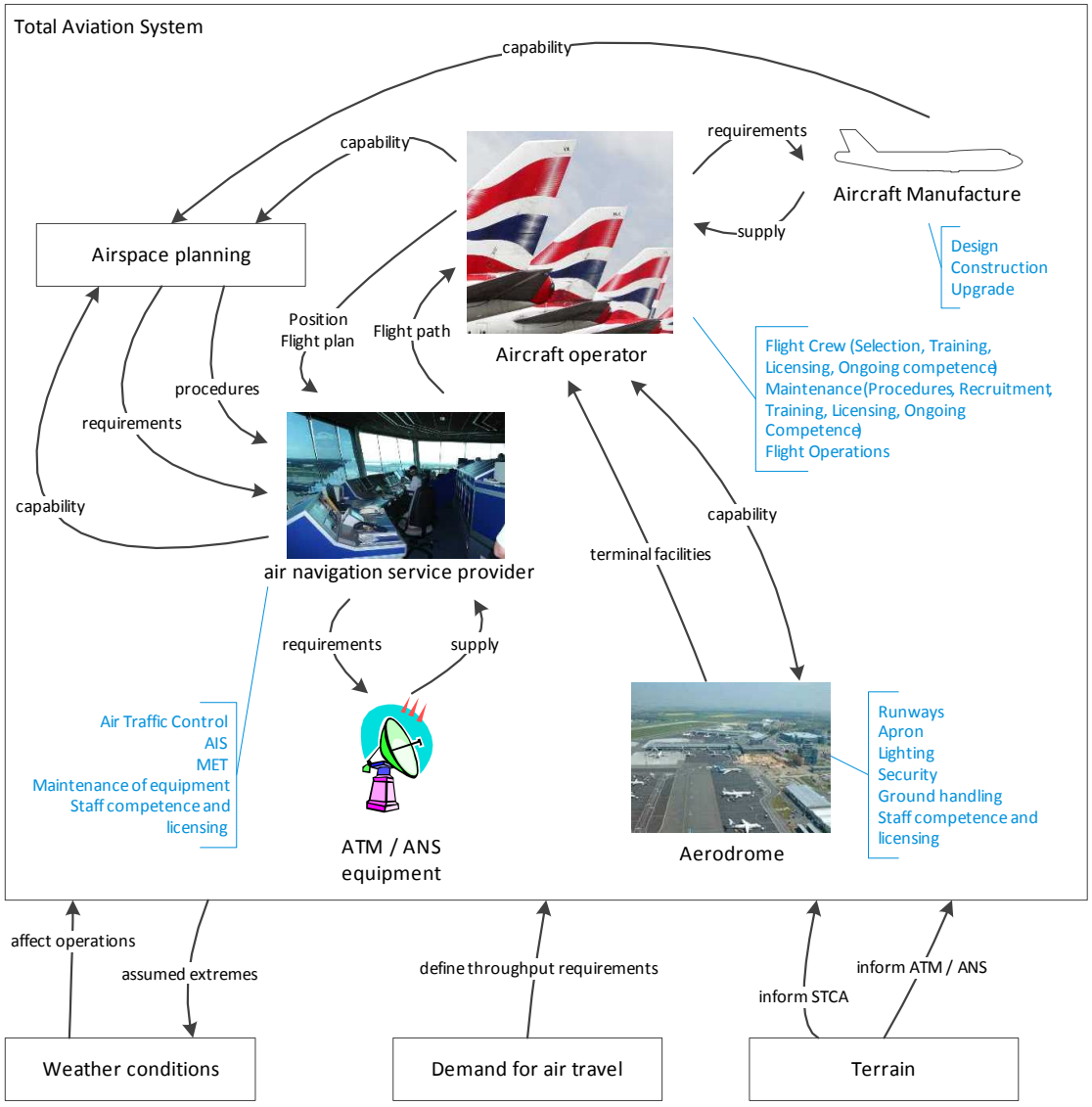

Figure 4: Illustration of total aviation system (TAS)

Appendix C Questionnaire Summary

C.1 Questionnaire development

The WP1.3 team developed a questionnaire to learn from the experience of people in the aviation industry – in particular their experiences (a) of integration between disciplines and (b) of shortcomings and bottlenecks.

The questions have been derived by considering the information needed to complete these streams of activity, as illustrated in Figure 5. The "leaf nodes" in Figure 5 represent the types of information needed, including reference to the section of the questionnaire where the relevant questions to elicit this information can be found. The leaf nodes also reference supporting information which provides context to the questions asked.

The focus of the questionnaire is to allow people to relate their experiences as this gives them the opportunity to express their concerns in a context which is familiar to them.

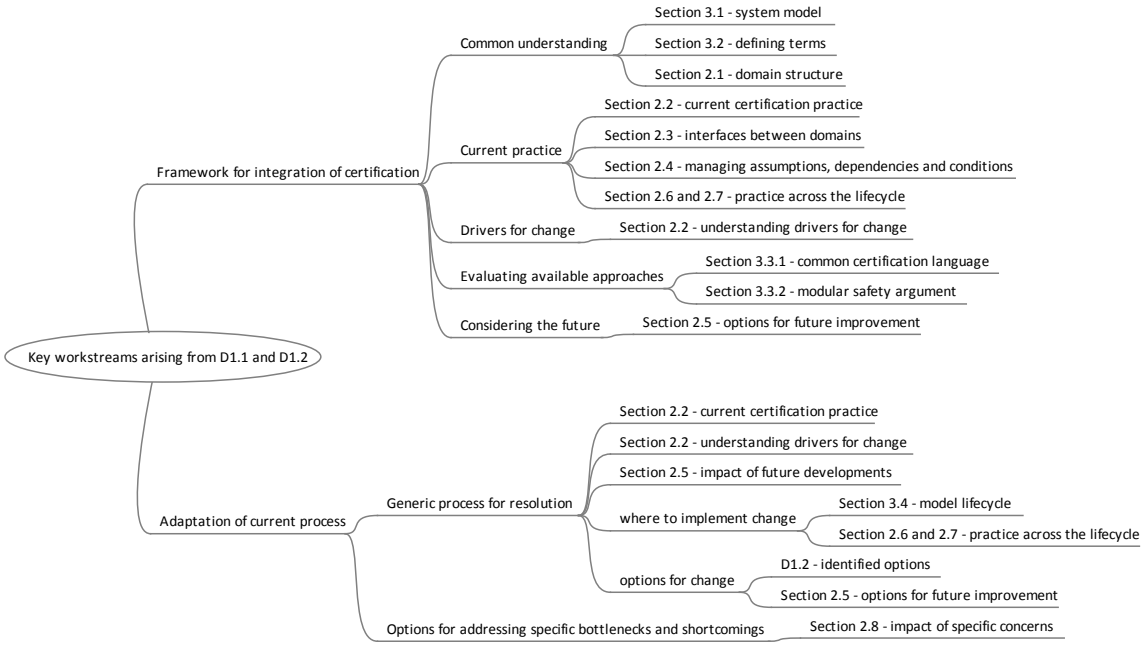

Figure 5: WP1.3 Approach and questionnaire structure
Section numbers refer to sections within the questionnaire document [10]

In summary of Figure 5, the purposes of the questionnaire were to:

- Gather information about the interaction between domains in the total aviation system.
- Gather information to help express how certification is achieved within the domains using a common language.
- Understand how certification activities in each domain differ / change across the implementation lifecycle.
- Understand the impact of bottlenecks and shortcomings on individual domains.

- Identify options for improving the integration between domains.
- Identify options for resolving bottlenecks and shortcomings.

C.2 Responses

Responses were received from the following;

Ref	Questionnaire response from a member of …
R1	Certiflyer (initial response)
R2	Cessna Aircraft Company
R3	Next Generation Aircraft BV
R4	Certiflyer
R5	EUROCAE

Table 3: List of the Questionnaire Responses

In reading across the responses twenty recurring themes emerged. These themes have been expressed in terms of either having a positive or negative influence on the reduction of bottlenecks or shortcomings. A positive effect is considered to be an improvement as it has the potential to reduce the number of bottlenecks or shortcomings, conversely a negative effect is considered to make matters worse as it has the potential to increases the number of bottlenecks or shortcomings.

The recurring themes and their assumed relationship to shortcomings and bottlenecks are illustrated in Figure 6.

Having established these themes the responses to each question were reviewed to see if they provided either support for a theme, an example for a theme or raise an issue related to a theme. In this way a rudimentary score was generated that highlights the main areas of concern as identified by the survey.

In addition to scoring the survey results the same scheme was used to score the input provided from meetings with the User Group ([47], [48]) and a separate meeting with EASA [46]. In this way one could check for any correlation and, more importantly, any contradiction in views.

The results have been plotted in Figure 7. The supporting detail of the responses and the user group input that contribute to the scoring has been extracted and appears in tabular form in section C.4.

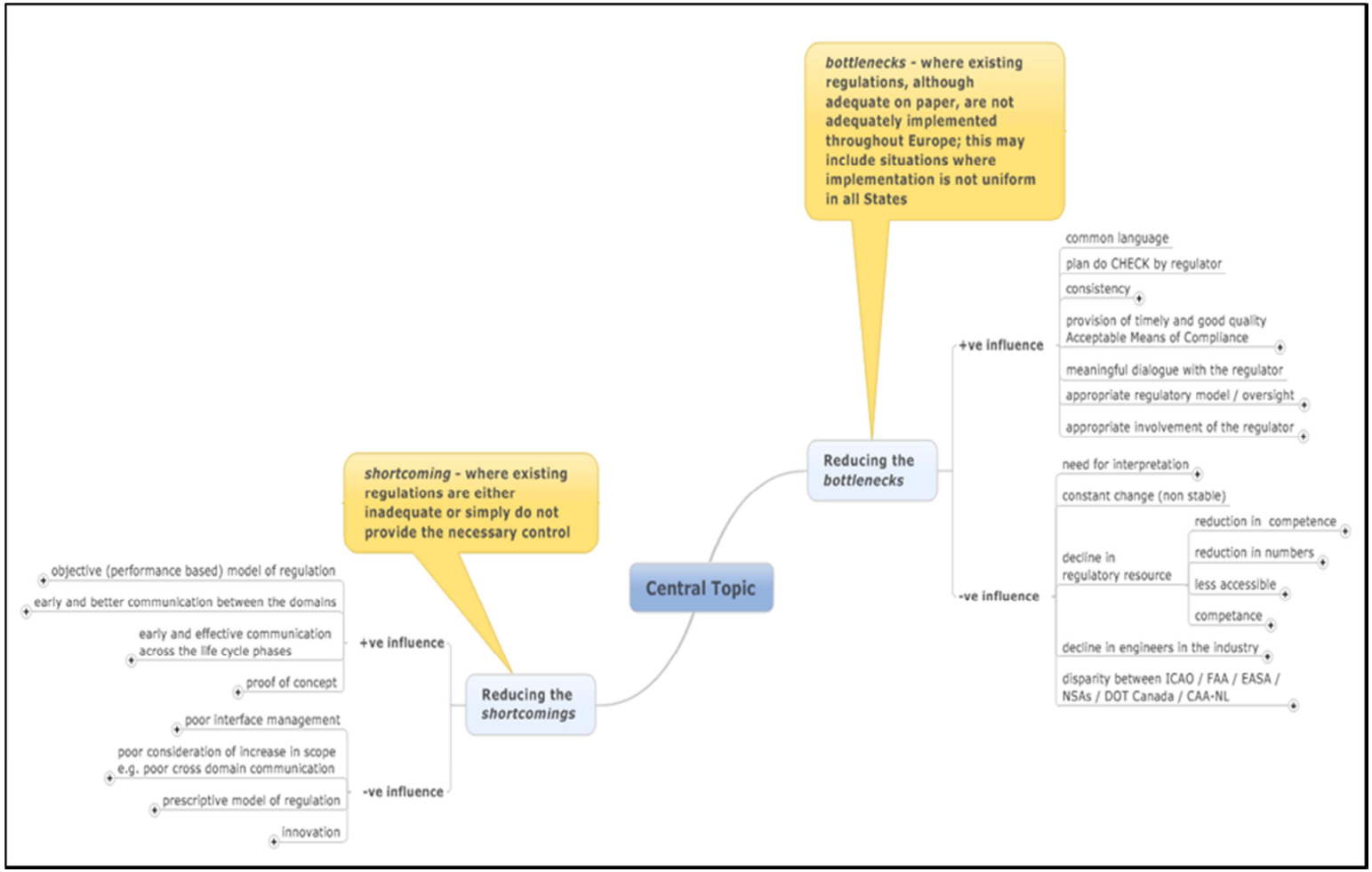

Figure 6: Recurring Themes and Their Relationship to Bottlenecks and Shortcomings

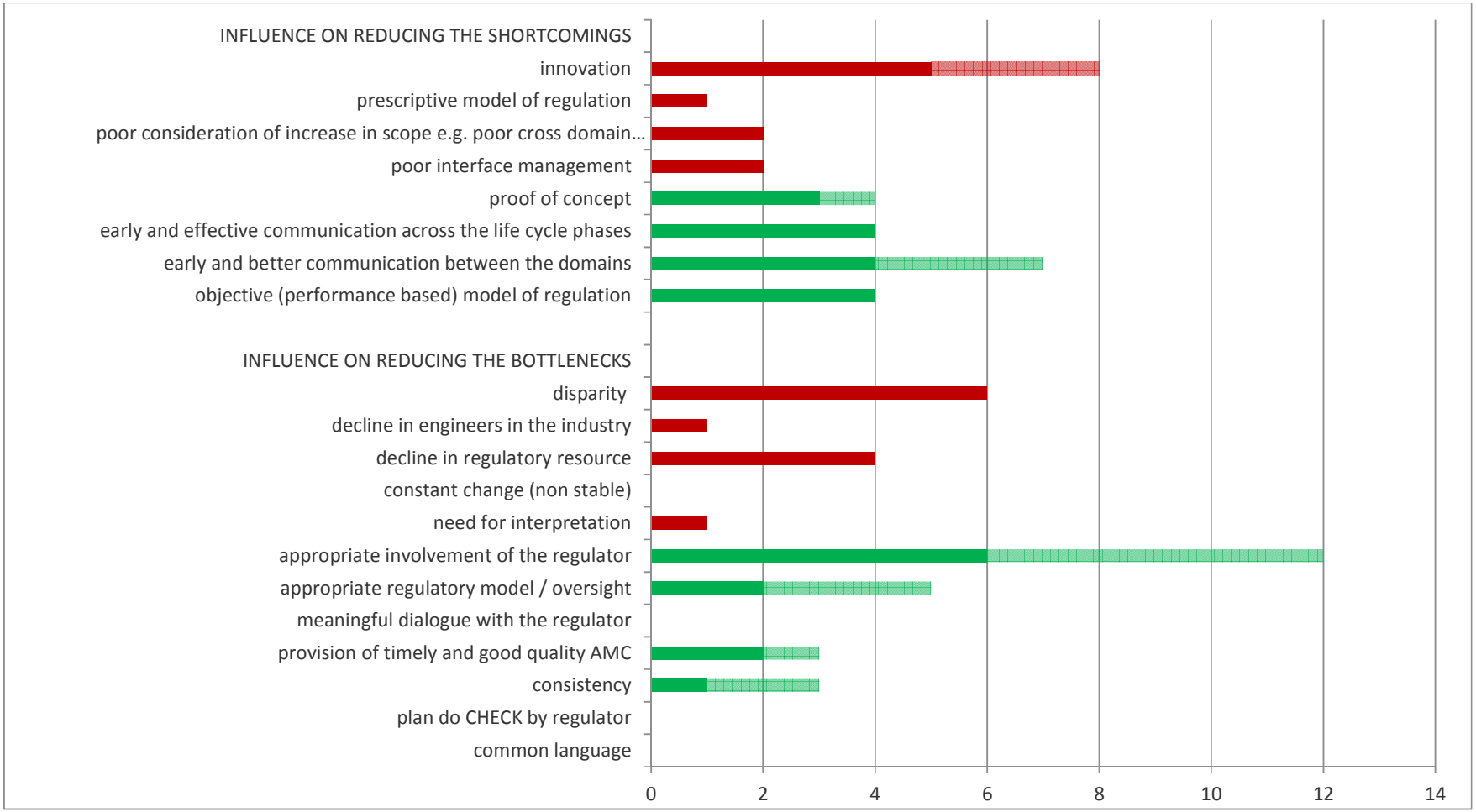

Figure 7: The Main Areas of Concern Identified by the Survey and the User Group

Key to Figure 7: Green indicates a positive influence on the reduction of shortcomings or *bottlenecks*. Red indicates a negative influence on the reduction of *shortcomings* or *bottlenecks.* The numbers indicate the number of responses related to the theme either by way of support, example or related issue. The darker shading indicates the input comes from the survey responses, the lighter shading indicates the input comes from the meetings with the User Group [47, 48] and EASA [46].

C.3 Analysis of Responses

In general the responses serve to confirm that this study is addressing the key areas of concern in the industry. The proposed certification approach addresses these key areas as described in section 3.5 in the body of the report.

Examples from the responses have been used to guide and illustrate the proposed certification approach presented in the body of this report, using indented paragraphs in this font. It may be beneficial to use the examples as additional test cases to test the degree to which the improvements that are to be proposed by ASCOS will address the problems / issues cited.

In addition the responses confirm the need for further integration between the domains, in particular where they identify "poor consideration of increase in scope" and "poor interface management" and show the need for "early and effective communication across the life cycle", "early and better communication between the domains" and "appropriate regulatory model / oversight".

C.4 Supporting Data

The following tables provide the detailed analysis of the questionnaire responses and of output from the User Group Workshops [47, 48] and the meeting with EASA [46].

The key to the tables is as follows:

> **Rx q.q supports:** ➔ Response 'x' (see Table 3) in response to question 'q.q' supports the theme in principle
>
> **Rx q.q example**: ➔ Response 'x' (see Table 3) in response to question 'q.q' provides an example that evidences the theme.
>
> **Rx q.q issue:** ➔ Response 'x' (see Table 3) in response to question 'q.q' raises an issue (or potential issue) related to the theme.
>
> **Rx q.q observation:** ➔ Response 'x' (see Table 3) in response to question 'q.q' makes an observation related to the theme.

A similar notation is used to refer to output from the meetings listed above.

C.4.1 Reducing the shortcomings (i.e. positive influence)

Theme	Support / Example / Issue
objective (performance based) model of regulation	**R1 4.1 supports:** ... there are benefits in a more Performance based Product Certification process in order to avoid delays in a certification process of technology that was not foreseen. This should not mean ... AMC should be discarded. A subtle change in certain areas (e.g. Avionics) could improve the effectiveness of the Certification process.
	R4 4.3 supports: progressive certification is already being done in Product Certification, but could be improved by more performance based certification
	R4 5.3 supports: reference further engine control automation, proposes more performance based requirements
	R4 5.4 example: reference further engine control automation, if there would be a generic requirement that would specify a safety level rather than a specific design solution (e.g. manual backup) the certification could be done without delay
early and better communication between the domains	**R4 2.10 supports:** reference FANS & ATM, improvement through earlier and better communication between the domains
	R4 5.3 supports: reference the fusion of FANS & ATN ... better coordination between domains
	R4 5.4 observation: reference the fusion of FANS & ATN ... within EASA there is now a good opportunity to lower the borders between the different domains and improve the coordination
	R5 Q8 supports: ... consider SESAR /ASBU timeline, consider not only Airworthiness but also Ground, Space & ATM segments
	[47] Action 3 supports: consider in the innovative approach of the certification process in WP1 not only “functional” point of view but also “data” point of view
	[46] Action 8 supports: Elaborate on Option 7 for adaptation of regulatory/certification process: start with a ‘bridge’, introduce a connection where a connection is needed. Take Operational Suitability Data (OSD) as an example for such a bridge: with assumptions about pilot competencies.
	[48] section 3.11 supports: For ASCOS it is not sufficient to only monitor the aircraft data. The total system needs to be monitored, including the interfaces between different parts of the system.
early and effective communication across the life cycle phases	**R4 3.6 example 1**
	R2 4.3 supports: progressive certification would eliminate issues with assumptions that drive future certification decisions
	R4 2.9 supports: reference FANS & ATM, it is a concern that the data link usage will be expanded ... to more demanding flight phases without improving the HMI. Now voice is primary, but for how long?
	R4 3.2 - 3.6 example: reference FANS & ATM, talks positively about the use of Design Assurance Levels, timely product certification and cooperation between product certification and operation ... leading to common review of design issues.
proof of concept	**R2 4.2 supports:** ... early development and full size proof of concepts are invaluable up front activity that could save time and money in the future

Theme	Support / Example / Issue
	R4 4.2 observation: contrary to the ATM domain, Product certification would never allow a proof of concept to be performed during commercial operations
	R4 4.2 example: two examples of proof of concept; evaluation of display features in the PFD in order to facilitate lower landing minima, the introduction of enhanced vision in order to facilitate lower minima
	[47] Action 2 supports: Consider in WP1 the outcomes of the SESAR 16.1.4 project "Proof of concept"

C.4.2 Increasing the shortcomings (i.e. negative influence)

Theme	Support / Example / Issue
poor interface management	**R4 2.5 example:** reference the fusion of FANS & ATN ... the impact on the flight deck was not foreseen by ATM regulation, a different perspective with respect to the use of the system existed between product certification and ATM.
	R3 2.7 example: there are no ground ice driven design requirements
poor consideration of increase in scope e.g. poor cross domain communication	**R4 2.5 example:** reference the fusion of FANS & ATN ... the impact on the flight deck was not foreseen by ATM regulation, a different perspective with respect to the use of the system existed between product certification and ATM.
	R5 Q9 example: too segmented, stake holders are not very much organised in an effective framework
prescriptive model of regulation	**R5 Q9 example:** certification effort / time not predictable
innovation	**R4 2.9 supports:** reference FANS & ATM, it is a concern that the data link usage will be expanded ... to more demanding flight phases without improving the HMI. Now voice is primary, but for how long?
	R4 2.9 example: reference FANS & ATM, reliance on automation, shift from clearance based to trajectory based ATC, increased complexities, technologies and procedures enabling reduced separation
	R4 2.9 example: reference FANS & ATM, 4D navigation with business trajectories and reduced separation
	R4 6.2 example: reference further engine control automation, Problem with older "Grandfathered" derivative designs. These aircraft are possibly not in compliance and will not be redesigned to comply. Economic burden.
	R5 Q9 example: innovation often dealt with in a reactive mode
	[46] Action 16 example: Certification is based on competences of the flight crew, But the more automation is used, the less experience is gained when manual take over is necessary. This should impact the certification of today. Encompass this in one of the options for adaptation of regulatory/certification process.
	[46] Action 21 example: Discuss the issue of suborbital flight with the ASCOS team and have a discussion with Jean Bruno Marciacq in more detail.
	[48] Action 1 example: Explore whether RPAS can be accommodated within one of the case studies. In case RPAS is included in the case studies, discuss the proposed change of the DoW with the EC.

C.4.3 Reducing the bottlenecks

Theme	Support / Example / Issue
common language	nil
plan do CHECK by regulator	nil
consistency	**R5 16 supports:** address the segmented approach with different methodologies to give consistency, lessons learned, recommendations.
	[46] Action 10 supports: Explore the possibility to identify – within ASCOS - whether there are overlaps, inconsistencies, redundant requirements or even contradictory requirements between different domains and remove these.
	[48] Action 20 supports: Construct a complete overview of all elements that do exist and do not exist (e.g. for ARP50 there is no ED, EUROCAE have developed guidelines for ground and ATM, these do not exist at ARP level).
provision of timely and good quality Acceptable Means of Compliance	**R4 2.6 example 1 supports:** certification ... on the aircraft side was heavily delayed as there was NO AMC developed at the mandatory introduction date [This example related to implementation of the European ATN system for which no AMC had been produced at the start of the project.]
	R4 2.6 example 1 supports: ... AMC was developed that was difficult to comply with ... delayed the introduction by approximately two years. Furthermore, most implementations were substandard needing AFM limitations on use. [This example related to implementation of the European ATN system for which no AMC had been produced at the start of the project.]
	[48] Action 8 supports: Provide guidance on writing safety arguments, including creating example arguments.
meaningful dialogue with the regulator	nil
appropriate regulatory model / oversight	**R3 2.11 supports:** make worldwide the ground de-icing of aircraft a maintenance task - thus being able to use certified de-icers with appropriate procedures and requirements
	R4 2.10 supports: reference further engine control automation, more high level requirements / more performance based requirements, that need not to be adapted every time there is new technology being developed by industry
	[46] Action 7 supports: Take constraints relating to public responsibility into account when analysing the different options for adaptation of regulatory/certification process.
	[46] Action 8 supports: Elaborate on Option 7 for adaptation of regulatory/certification process: start with a 'bridge', introduce a connection where a connection is needed. Take Operational Suitability Data (OSD) as an example for such a bridge: with assumptions about pilot competencies.
	[46] Action 12 supports: Consider an additional option for adaptation of regulatory/certification process: a combination of product and organization certification
appropriate involvement of	**[-] no reference:** guidance on the trigger

Theme	Support / Example / Issue
the regulator	**R2 4.3 supports:** progressive certification would eliminate issues with assumptions that drive future certification decisions
	R3 3.0 issue: EASA not living round the corner, need to apply for technical opinion, costs money
	R3 4.2 example: use of red label s/w on revenue flights agreed with Swiss FOCA - resulting in a greatly improved FMS VNAV function
	R5 Q8 supports: streamlined approach using industry standards early in the regulatory framework
	R4 2.10 supports: reference FANS & ATM, improvement through earlier publication of requirements

C.4.4 Increasing the *bottlenecks*

Theme	Support / Example / Issue
need for interpretation	**R3 3.6 supports**:
constant change (non stable)	
decline in regulatory resource	reduction in competence - **R3 3.6 example:** Good relations with authority specialists is of paramount important, this is because it's not the requirements that are the issue but their interpretation. Increasingly, the competence level of authority specialists is deteriorating. This is not helped by the de-commissioning of all JAA study groups. The result of this is that the know-why of the requirements - why are they as they are - is increasingly getting lost.
	reduction in numbers - **R4 2.5 example:** reference further engine control automation, a more performance based requirement would have solved the problem but the change of this requirement to a more performance based requirement was put on hold due to the lack of personnel at the Authority
	less accessible - **R3 3.0 issue:** EASA not living round the corner, need to apply for technical opinion, costs money
	competence - **R4 2.7 example:** reference further engine control automation, the dependency is that of the manufacturer on the Authority. And the Authority on its limited capability to change the requirements.
decline in engineers in the industry	**R3 3.7 example:** Accident investigation and resolution increasingly difficult because suppliers do not have their original engineers
disparity between ICAO / FAA / EASA / NSAs / DOT Canada / CAA-NL	**R3 2.5 example:** F100 de-icing
	R3 2.6 example: de-icing of aircraft cited as a maintenance activity in the USA versus airfield activity elsewhere
	R3 2.8 example: Leading edge heating was mandated by the FAA in 1999, and by CAA-NL in 2009
	R4 2.8 example: reference further engine control automation, there is a probability that one authority is able to accept the change by issuing an Equivalent Safety Finding as a temporary band aid, but it is unclear if this will be accepted by other authorities.

Appendix D Template Arguments

This Appendix presents template arguments for common approaches. These are presented at a high level to illustrate the approach and will be developed during the case studies. In addition, further templates will be developed.

Figure 8 presents a template argument for the safety of flight operations. This is divided according to the three stages followed by the UK CAA: operators are first certified for general flight operations, then they are licensed to operate specific aircraft and routes. Finally the CAA provides oversight of the operations to ensure continued compliance. The process of developing the template this far has identified a number of areas which need careful consideration in the development of arguments for specific scenarios.

- What form of safety targets are appropriate for demonstrating safety of flight operations?
- How can we demonstrate that the certification and licensing requirements define the appropriate safety criteria?

Figure 9 presents a template argument for the **safety of a proof of concept demonstration**. This is divided into an argument in each of three domains (assuming that these are the (only) domains affected by this proof of concept – this is captured as assumption A001). This argument follows the approach presented in the SESAR guidance material ([18] section 6.4). When the argument is developed for a specific case, the lower levels would show some or all of the features identified in that material, including:

- stopping criteria for the demonstration (i.e. if continuous monitoring showed an unacceptable impact on safety risk);
- human factors analysis, including fall back and contingency procedures and training needs analysis.

Figure 10 presents the top level of a template for a **pure compliance-based argument**. The key feature to note here is the need to demonstrate

a. that the standards set sufficient requirements (Cl 1);
b. that the environment assumed by the standards is understood, and assessed for any differences from the environment in which the change is being made.

This need for rigorous assessment of differences is recognised in the rail industry: the concept of cross-acceptance (see Appendix E.2.3), which provides for acceptance of a product to be transferred between railway authorities, requires formal assessment (and mitigation) of differences in product, processes and environment.

Many arguments are, in fact, a combination of compliance-based and performance-based approaches: even where the predominant approach is performance-based, elements of a compliance based argument (which could use this template) are often still required.

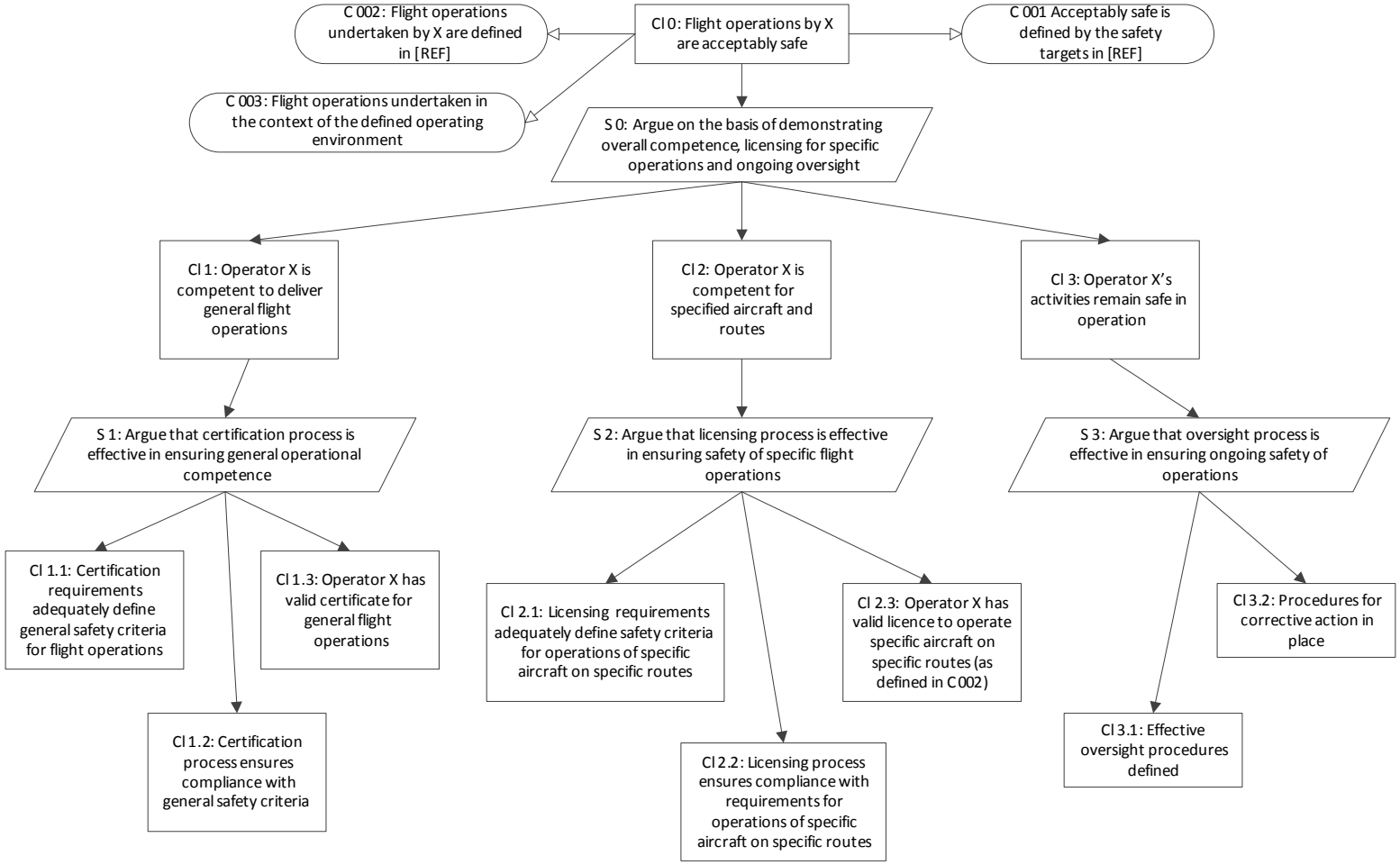

Figure 8: Template argument (high level) for flight operations

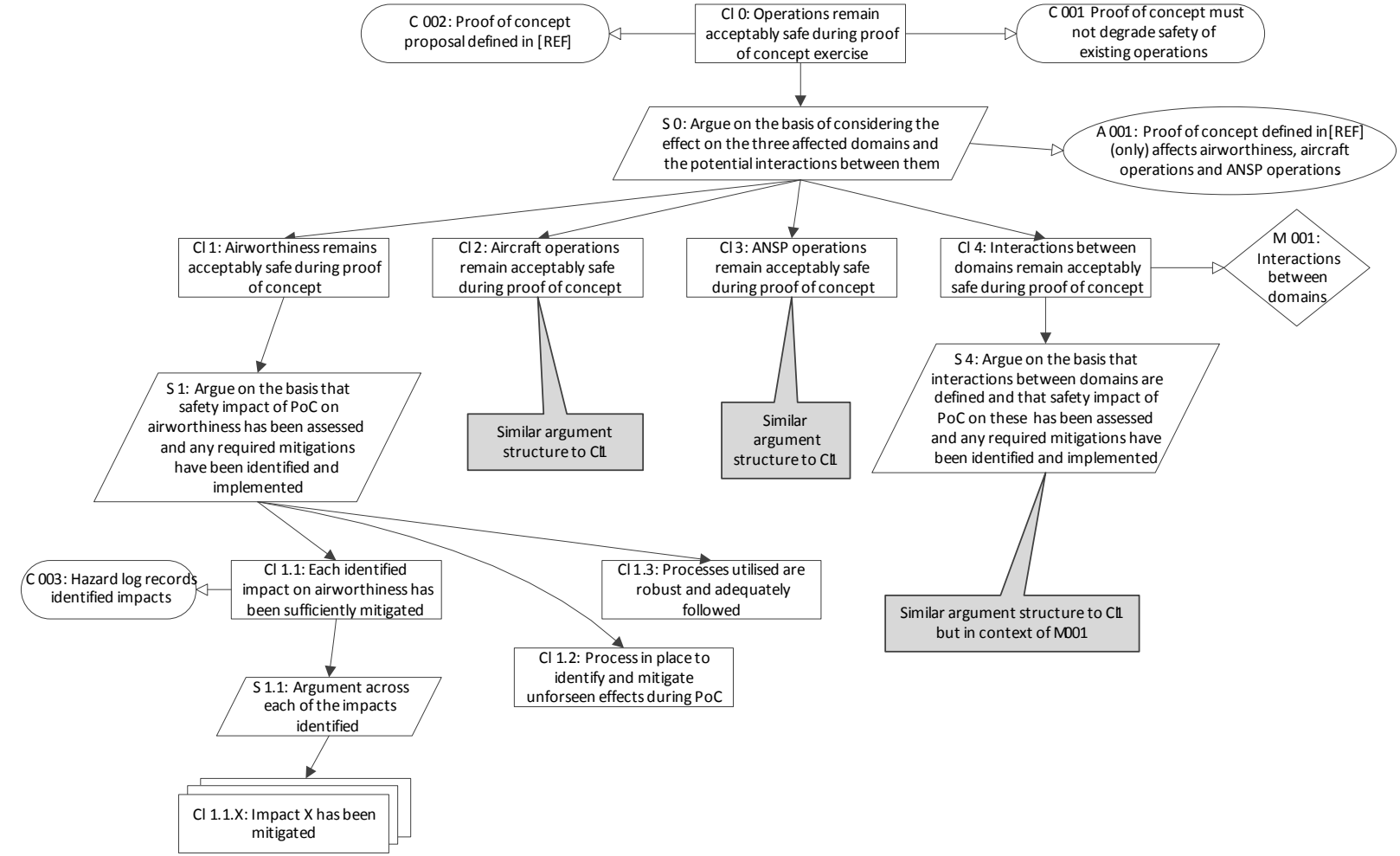

Figure 9: Template argument (high level) for safety of proof of concept

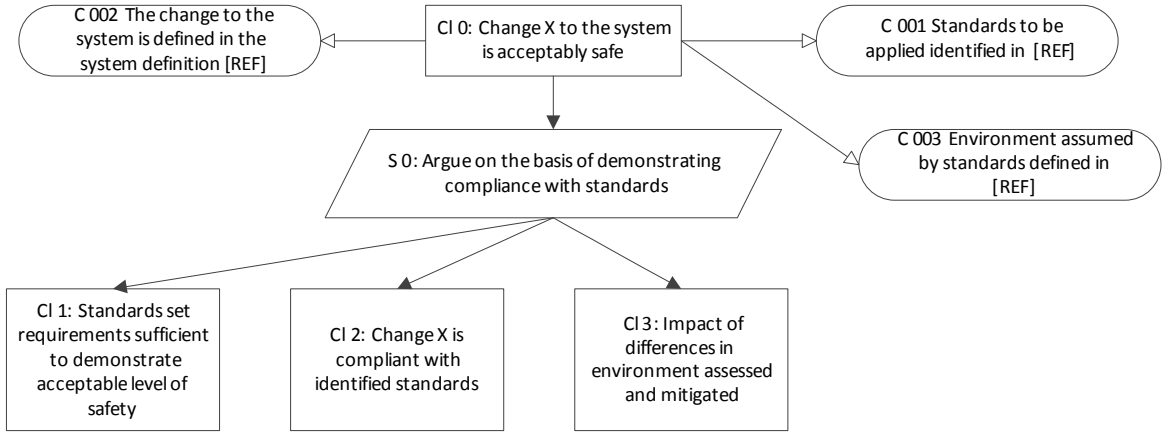

Figure 10: Template (high level) for pure compliance-based argument

Appendix E Related Approaches Across Industry

E.1 OPENCOSS

Another ongoing EU research project (OPENCOSS - Open Platform for Evolutionary Certification of Safety-Critical Systems), working across industry sectors, has undertaken as review of the most promising of these concepts and tools[30] within realm of product certification: we are drawing on some of these in this work package and applying them to the ASCOS scope of work.

E.1.1 Common Certification Language

One of the goals of OPENCOSS (WP4) is to define a common framework for describing certification arguments. Use of a common framework helps to ensure common understanding across different domains. This helps to ensure that arguments are compatible and that full coverage of issues is achieved.

This common framework is described in the OPENCOSS documents as a Common Certification Language (CCL). However, the primary aim is not to develop a new means of representing safety arguments; instead the aim is to provide a standardised lexicon in which such arguments can be expressed. The CCL is intended to cover:

- Description of safety cases (and safety arguments).
- Characterisation of evidence (i.e. evidence metadata) supporting the arguments – this assists in evaluation and reuse of evidence in contexts other than that for which it was first developed.
- Description of compliance management.

The CCL has not yet been published; the deliverables which have been published:

- Cover the concepts and describe current practice (D4.1) [4]; and
- Define requirements for the CCL (D4.2) [5].

In addition, OPENCOSS has published high level requirements (D2.2) [3] which include a glossary of concept definitions.

We have drawn on the concepts within D4.1, in conjunction with the glossary in D2.2, to develop our definitions for ASCOS WP1.3; these are presented in Appendix A for ease of reference. (It will be useful to compare these definitions with the CCL when it is published.)

E.1.2 Compositional Certification

Another goal of OPENCOSS (WP5) is to consider how certification arguments can be structured to support re-use of elements of the argument (e.g. pertaining to a particular service or item of equipment) without the need to redevelop the argument from scratch. The key principle is to split the argument into well-defined modules, with defined interfaces; this is analogous to similar principles in software and system design. As with

[30] OPENCOSS is purely looking at approaches to certification. It does not consider safety assessment methodologies – these will be separately considered by ASCOS, particularly in WP3.

WP4, this work package is not complete, but the concepts of compositional certification are well presented in D5.1 [6]. We have drawn on the concepts in providing our overview of certification argument structure.

E.2 Rail Sector

E.2.1 Common Safety Methods (CSMs)

The European rail sector defines methods called "Common Safety Methods" to standardise approaches across Europe. CSMs are defined for:

- (Safety) Risk Evaluation and Assessment (EC 352/2009[31]) – this essentially prescribes the identification and management of hazards for all "significant" changes to the railway, and sets out three options for demonstrating that the risk from those hazards is acceptable. The three options are: appeal to standards, comparison against an existing system, detailed risk analysis.
- Monitoring (EC 1078/2012) – defines how operating organisations should monitor their compliance with their safety management system
- Conformity Assessment (EC 1158/2010 and 1169/2010) – defines how (rail) competent authorities assess the SMS (proposed by) an operating organisation
- Supervision (EC 1077/2012) – defines how competent authorities supervise the ongoing implementation of an operating organisation's SMS.

The introduction of CSMs is an ongoing process and they are still evolving as the sector learns from experience.

E.2.2 Acceptance Process in UK Rail Industry

Network Rail[32] requires any products deployed on the UK main line railway are formally accepted before deployment.

NR encourages applicants to commence the acceptance process in good time before intended deployment and they use the concept of technology readiness levels (TRLs) to guide when it is appropriate to commence the process. This gives the opportunity for NR to influence development at an appropriate stage and prevents the acceptance process from becoming a bottleneck before introduction of the equipment.

The acceptance process is summarised on the Network Rail website [22**]** which also contains a link to a five page guide to the process.

E.2.3 Cross Acceptance

As noted above, formal acceptance of railway products by the railway infrastructure manager (IM). As part of the European interoperability directives, an IM is required to accept products which have been accepted by

[31] Updated CSM published as EC 402/2013, coming into force in 2015
[32] Network Rail is the main line rail infrastructure manager in the UK, responsible for operating and maintaining the fixed railway infrastructure for the main line network, but not for the London Underground and other light rail systems.

another IM (e.g. in another State) without repeating full assessment of the product. However, it is recognised that differences can invalidate acceptance, so a guidance note [28] has been published to establish a framework for assessment of these differences. This guidance note elaborates on 7 key principles for establishing a cross-acceptance argument.

a) Establish a credible case for the native (baseline) application
b) Specify the target environment and application
c) Identify the key differences between the target and native cases
d) Specify the technical, operational and procedural adaptations required to cater for the differences
e) Assess the risks arising from the differences
f) Produce a credible case for the adaptations adequately controlling the risks arising from the differences
g) Develop a generic or specific cross-acceptance case

E.2.4 Generic Safety Cases - Modularisation of Safety Arguments

The CENELEC standard EN 50129 governs the construction of safety cases in the rail industry. (In particular, safety cases for railway signalling systems.) This standard introduces three classes of safety case:

- Generic Product Safety Case (GPSC) makes the case that a generic product safely meets a defined set of requirements, with the intention that these requirements are representative of the expected / intended use of the product;
- Generic Application Safety Case (GASC) makes the case that a generic product (or system[33]) can be safely applied in a defined environment, but without being specific to a particular location – the intention is that this GASC can then be used to support application of multiple instances of the product;
- Specific Application Safety Case (SASC) makes the case for a specific application of the product (or system).

Generic safety cases are based on assumptions about the use of the system and the environment in which it will be used and demonstrate that the product or system will be safe as long as these assumptions are valid and any conditions on the use of the product or system are met.

This concept supports the modularisation of safety arguments by making a reusable argument. The skill in developing such cases is in ensuring that the assumptions are broad enough to be useful, while not being so broad that the system cannot be demonstrated to be safe for the full range of the allowed environment.

A key element of this approach is the generation of Safety Related Application Conditions (SRACs) within the lower level safety cases which define the constraints and limitations on the use of the product; one important

[33] Often the GASC and SASC make the argument for a combination of products; indeed one reason to generate a GASC is to show the safe combination of products (each supported by a GPSC) into a system.

activity for the higher level safety cases is to demonstrate that these SRACs are satisfied. This approach is documented in EN50129 [7].

Appendix F Fallacious arguments reading list

This appendix presents a list of references to research and analysis into fallacious arguments.

- W.S. Greenwell, J.C. Knight, C.M. Holloway, J. Pease: “A Taxonomy of Fallacies in System Safety Arguments”, International System Safety Conference 2006 (ISSC 06) [36]
- S. Toulmin, R. Rieke and A. Janik, "An Introduction to Reasoning", Macmillan Publishing, New York, 1979 [51]
- D.N. Walton, "Reasoned Use of Expertise in Argumentation", Argumentation Vol. 3, pp.59-73, 1989 [52]
- D.N. Walton, "Argumentation and Theory of Evidence", in New Trends in Criminal Investigation and Evidence, vol. 2, pp.711-732, 2000 [53]
- Hahn and Oaksford, A Bayesian approach to informal argument fallacies, Synthese (2006) 152 pp 207 – 236 [54]

Appendix G Key to Goal Structuring Notation

The safety argument which forms the basis of this Safety Case is presented in Goal Structuring Notation (GSN). See 'Goal Structuring Notation Community Standard' [14] for an overview of the notation and its rationale. A key to the symbols used in this document is given in Figure 11 and Figure 12 below.

Elements of the argument are numbered uniquely and hierarchically. Elements providing context to goals and strategies are numbered using the number of that element, plus "-n" to provide unique identification.

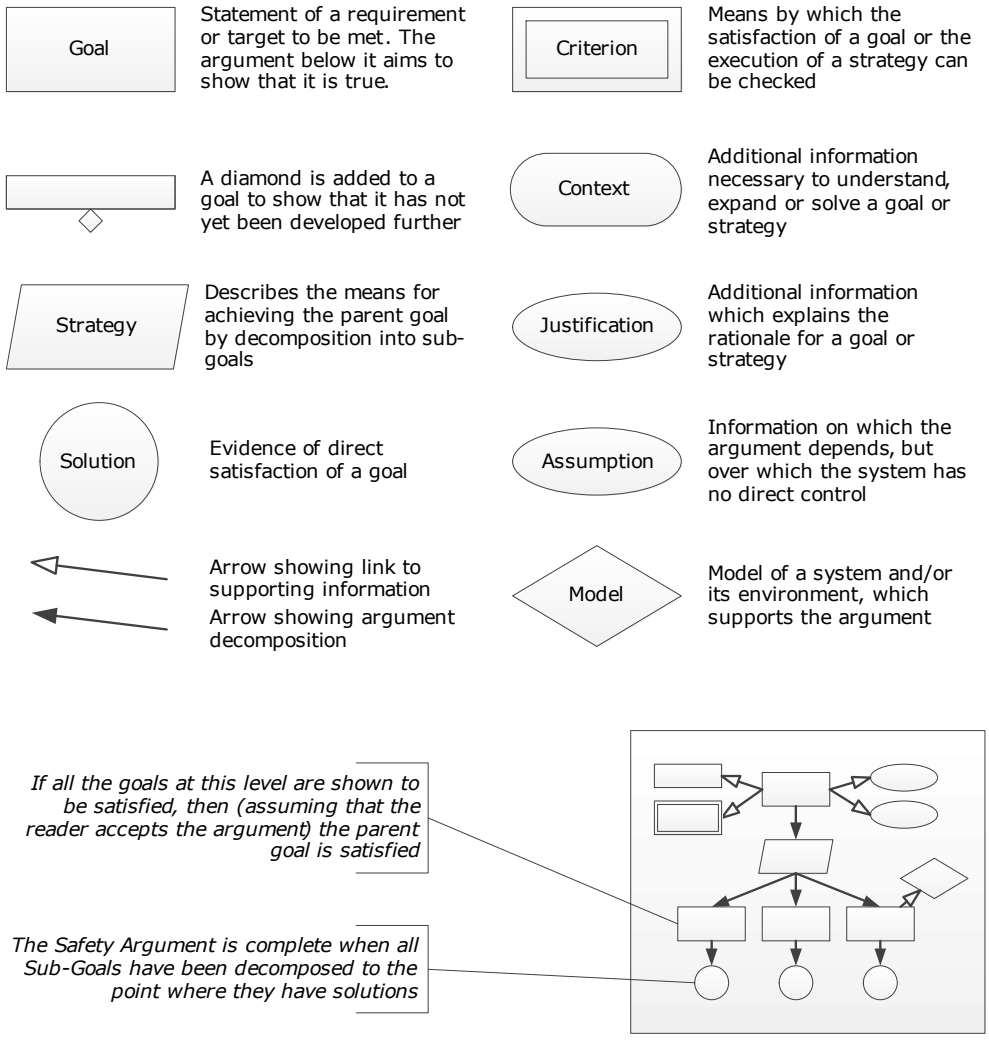

Figure 11: Key to basic GSN Symbols

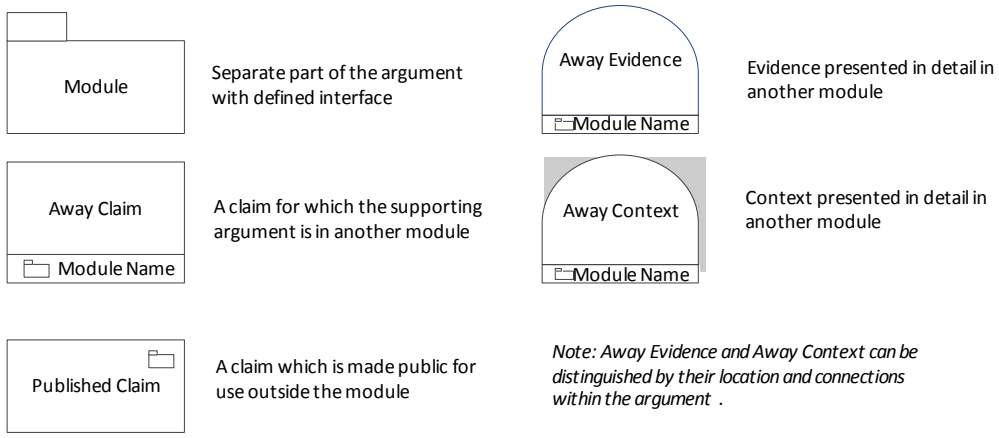

Figure 12: Key to GSN Symbols for Modular Arguments