Consolidated New Approval Method

S. Bull, J. Latham, M. Shuker (Ebeni), S. Barker, A. Eaton (CAA UK), B. Dziugieł, M. Maczka (IoA)

This document presents the ASCOS Method for approval of changes to the Total Aviation System and is especially relevant for changes which challenge existing approval approaches, either because of novel technologies or because they impact multiple approval domains. The ASCOS Method forms a framework within which existing approaches can be adapted or augmented as required – thus maximising efficiency in demonstrating the safety of the proposed change. This document is relevant to all who are involved in the approval of such changes, including applicants and approving authorities.

Coordinator	L.J.P. Speijker (NLR)
Work Package Manager	B. Pauly (Thales Air Systems)

Grant Agreement No.	314299
Document Identification	D1.5
Status	Approved
Version	1.1
Date of Issue	24-09-2015
Classification	Public

This page is intentionally left blank

Executive Summary

Aviation is undergoing significant and fundamental change. The dramatic increase in traffic, driven by increased demand, along with environmental requirements and other pressures, is driving the introduction of novel concepts and technologies and increasing the integration between the different domains of the total aviation system (TAS). When coupled with differences in underlying approval approaches between domains, these changes make it imperative to streamline the approval processes used across the industry. The ASCOS Consortium proposes a consolidated approval method for use across all domains, building on existing good practices, guiding applicants and authorities to consider the full impact of a change on the TAS. This is a requirement of recent EASA rules, and ensures that interactions between parts of the TAS are fully managed. The ASCOS Method is presented in the form of guidance to support the current EASA rulemaking programme without requiring further rule changes.

consolidated approval method addressing innovation and integration

The ASCOS Method focuses on establishing an approval path for a change to the TAS, using existing approaches which are adapted and augmented only when necessary. (This may be to accommodate innovation, to ensure interfaces are managed or simply to streamline the process.) The ASCOS Method provides a framework for development of such adaptations, which provides support throughout the lifecycle, starting with identification of the concept and establishing its viability, through development and implementation into operation and sustainment. The activities do not depend on a particular lifecycle being followed. The ASCOS Method is not just applicable to certification; it is also applicable to more general approvals.

integrated into lifecycle

The ASCOS Method includes the development of a logical justification, in the form of a safety argument, that the proposed change achieves the required level of safety. The safety argument is presented as a hierarchical set of claims, supported by evidence, and is developed to consider all aspects of the TAS affected by the change. The ASCOS Method divides the safety argument into modules aligned to the domains of the TAS, which can be developed and decomposed further separately within the domains. Assurance contracts are used to manage the dependencies between modules. The modules allow the safety argument to be structured in a way which integrates with the existing structure and hierarchy of the organisations within the TAS. The concept of an argument architect is introduced to support management of the safety argument and of assurance contracts. The structure of the safety argument can be presented in a graphical form to aid understanding, although it is always supported by text to explain what is being claimed.

logical justification focussed on safety across the Total Aviation System

The ASCOS Method recognises the significant underlying differences in approach between domains, including levels of safety, assessment methods and terminology; sometimes different domains give significantly different meanings to the same term. Differences between domains are understandable given the structure and history of different parts of the TAS, but careful consideration is needed in building an integrated method. The

recognises and reconciles differences between domains

method does not in itself mandate how safety targets for a change should be established, but recognises that the current high level of safety must be maintained.

The ASCOS Method is capable of addressing a wide range of changes to the TAS, including introduction of new operational concepts, new organisations, revised processes or new aviation products that affect operation. It is applicable across all domains (including aircraft, ATM, aerodrome, crew training, maintenance activities and airspace structure) with the greatest benefits obtained for changes which span multiple domains.

flexible, embracing innovation

Novelty and innovation require flexibility and thus the ASCOS Method provides a framework within which new approaches (e.g. goal-based justifications where the change is beyond what is envisaged by existing standards) can be adopted where necessary, while retaining or adapting existing approaches where appropriate.

guidance to tailor the approach

Guidance is provided to show how the method should be adapted according to the needs of an individual change. This recognises that although the overall concept can be applied to any change, the detailed method will vary widely depending on the particular change to be made – for example, the safety argument for introduction of a new equipment item on an aircraft will be very different from the safety argument for a change to the arrivals concept at a particular aerodrome.

drawing maximum benefit from ASCOS

The ASCOS Method has been developed from the proposal made in ASCOS D1.3, taking input from participants in the ASCOS programme, including the safety monitoring and modelling tools, the case studies and validation exercises. The safety argument concepts are developed from earlier work by SESAR and EUROCONTROL. The ASCOS Method takes the aviation community closer to a fully optimised approach to the approval of change.

In summary, the ASCOS Method responds to the pressures in the aviation industry which are driving innovation and increased integration between domains and therefore making it imperative to streamline approval processes. The ASCOS Method integrates with the lifecycle of a change, from concept through into operational service, introducing activities which lead to building a safety argument supporting the application for approval. The proposed method considers the full impact of the change, and recognises and manages the interaction between domains. The method is also flexible to embrace innovation while encompassing existing established processes wherever appropriate.

Finally, further opportunities for improvement and refinement of the ASCOS Method have been identified. However, the greatest opportunity for improvement will come from application of the ASCOS Method. The ASCOS Consortium commends this ASCOS Method to EASA for adoption as a means of establishing approval for changes to the TAS within Europe.

Document Change Log

Version	Author(s)	Date	Affected Sections	Description of Change
1.0	Stephen Bull et al.	15-09-2015	All	Version for approval by PMT
1.1	Stephen Bull	24-09-2015	Front Page, 8.3.8	Minor updates following review

Review and Approval of the Document

Organisation Responsible for Review	Name of person reviewing the document	Date
NLR	A.L.C. Roelen, P.J. van der Geest, M. Stuip	07-09-2015
CertiFlyer	G. Temme	07-09-2015
Ebeni	A. Simpson	07-09-2015
Institute of Aviation	K. Piwek, A. Iwaniuk	07-09-2015
Thales Air Systems SA	F. Kaakai, B. Pauly, F. Orlandi	07-09-2015
CAA UK	T. Longhurst	07-09-2015
TU Delft	R. Curran, H. Udluft	07-09-2015
Organisation Responsible for Approval	Name of person approving the document	Date
Thales Air Systems SA	F. Kaakai, B. Pauly	11-09-2015
NLR	L.J.P. Speijker	25-09-2015

Document Distribution

Organisation	Names
European Commission	M. Kyriakopoulos
NLR	L. Speijker, A. Rutten, M.A. Piers, P. van der Geest, A. Roelen, J. Verstraeten, A. Balk, E. van de Sluis, M. Stuip, G. van Baren
Thales Air Systems GmbH	G. Schichtel, J.-M. Kraus
Thales Air Systems SA	B. Pauly, F. Kaakai
Airbus Defence and Space APSYS	S. Bravo Muñoz, J.P. Heckmann, M. Feuvrier
Civil Aviation Authority UK	L. Young, A. Eaton, S. Barker, T. Longhurst
ISDEFE	I. Etxebarria, C. Regidor Gil
CertiFlyer	G. Temme, M. Heiligers
Avanssa	N. Aghdassi
Ebeni	A. Simpson, J. Denness, S. Bull, Latham
Deep Blue	L. Save, S. Rozzi
JRC	W. Post
JPM	J. P. Magny
TU Delft	R. Curran, H. Udluft, P.C. Roling
Institute of Aviation	K. Piwek, A. Iwaniuk, B. Dziugiel
CAO	A. Ortyl, R. Zielinski
EASA	E. Isambert
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

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 The ASCOS Project

1.1.1 Introduction to ASCOS

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, suggest the need 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 [1] and Flight Path 2050 [2] goals.

ASCOS is delivered by a consortium of organisations involved in the European aviation industry and supported by a wide ranging User Group providing input and review.

1.1.2 Objective for the ASCOS Project

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 changes to the aviation system (in particular 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.

1.1.3 Structure of the ASCOS Project

The ASCOS Project was structured into six main work packages:

- WP1: Certification Process – Development of safety based certification process adaptations based on analysis of existing certification and rulemaking process and evaluation of different possible new approaches
- WP2: Continuous Safety Monitoring – Development of a methodology and supporting tools for multi-stakeholder continuous safety monitoring, using a baseline risk picture for all parts of the total aviation system
- WP3: Safety Risk Management – Development of a total aviation system safety assessment methodology, with supporting safety based design systems and tools, for handling of current, emerging and future risks
- WP4: Certification Case Studies – Application of the new certification approach and supporting safety based design systems and tools in the selected example case studies
- WP5: Validation – Validation of the new certification approach and the supporting methods and tools

- WP6: Dissemination and Exploitation – Dissemination to ensure that results are correctly understood and exploited to the maximum extent

The project is also supported by a seventh work package for project management.

The relationships between these work packages are depicted in Figure 1.

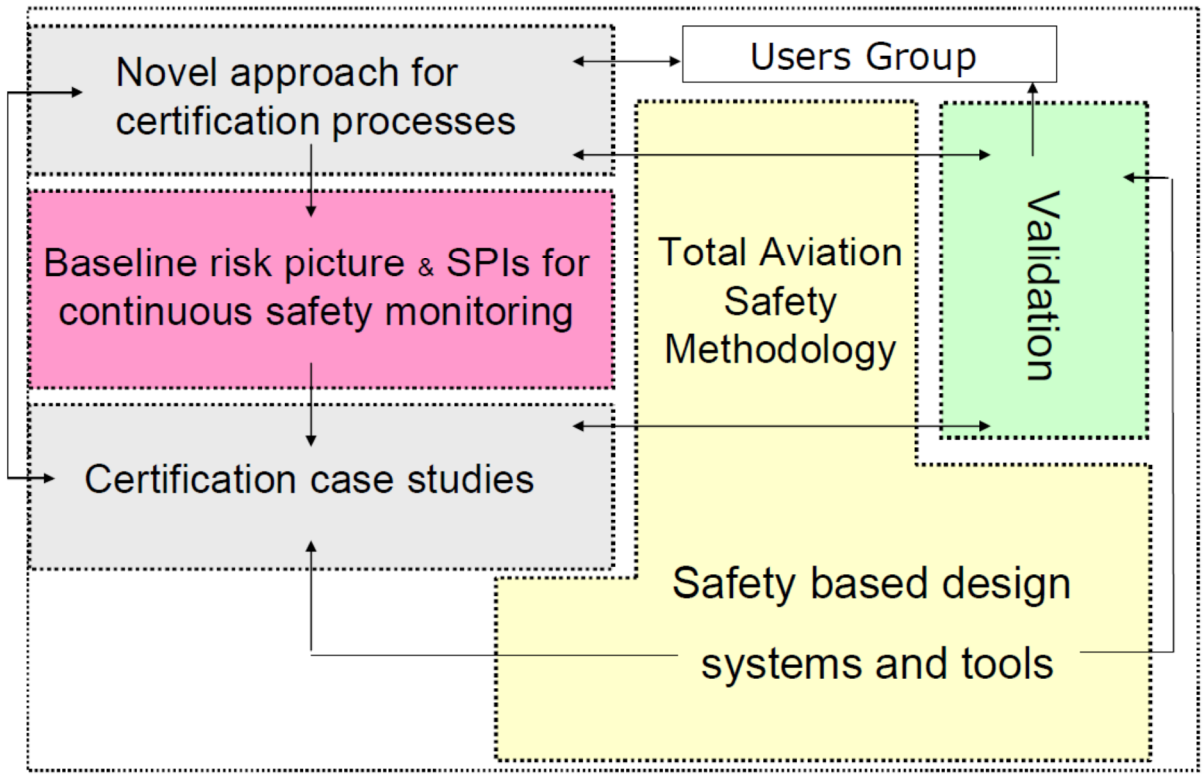

Figure 1: Relationships between ASCOS work packages

1.2 ASCOS Certification Process Work Package

The aim of the certification process work package (WP1) is "to develop safety based certification process adaptations based on analysis of existing certification and rulemaking process and evaluation of different possible new approaches." The ASCOS remit (Description of Work) also calls for the proposed certification adaptations to deliver:

- 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

The initial activities in this work package reviewed current regulations and the degree to which these regulations are implemented within the aviation community, examined accident statistics and trends and

identified potential bottlenecks and shortcomings in the current certification processes. Eight possible options for improvement were identified and evaluated, and four were chosen as a basis for further work:

- 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

In addition, the following principles were identified which were used to govern the further development of the ASCOS Method:

- 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 environment, to ensure that all key safety issues are properly addressed and not lost at interfaces
- Allow, within each domain, the new method to evolve from current approaches 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 existing 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, in particular between aircraft certification and Air Traffic Management
- 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

The above options and principles were used to develop a proposed certification approach, which is presented in ASCOS deliverable D1.3 [3]. (Assessment of the method against these principles is presented in section 8.2.)
Note: this document supersedes D1.3.

1.3 Objective and Scope of D1.5 (this document)

This document presents a consolidated method for the approval of a *change*[1] to the *Total Aviation System (TAS)*, herein referred to as the ASCOS Method. The ASCOS Method has been generated by refining the D1.3 approach following feedback from the case studies (WP4) and from validation exercises (WP5). This document represents the final output from the ASCOS project in respect of a consolidated *approval*[2] method.

The ASCOS Method, is a framework within which any *change* to the *Total Aviation System* (*TAS*) can be assessed to determine whether it achieves its goals in respect of the safety of the *TAS*. A *change* may range from upgrading an obsolete product, through introduction of a new product (including new aircraft) or process, to introduction of a novel operational concept (such as self-separation). In many cases, changes can be demonstrated to be safe using existing processes, and in these cases the framework provides only limited benefits. However, where (as is happening increasingly) changes span multiple *domains* of the *TAS* and / or introduce technologies or concepts not envisaged by existing standards, the framework provided by the ASCOS Method allows existing approaches to be integrated with new approaches and thus ensures that the full impact of the changes across the *TAS* is suitably addressed.

The ASCOS Method also:

- allows existing approaches to remain in use and provides guidance on how to evaluate ways in which these approaches may need to be extended or augmented to address the challenges of a particular *change*
- provides, in the safety arguments for individual changes, building blocks towards a safety argument for the *TAS* itself
- complements the work done elsewhere in the ASCOS Project ([5]) in proposing improvements to existing standards.

It is difficult to introduce the flexibility to accommodate innovation and to address *changes* which span the *TAS* (the second and third objectives above) without having a negative impact on the cost and efficiency of the *approval* process, at least in the short term. In addition, the innovations envisaged within aviation may also drive up the scale and complexity of the safety assurance required, having a further negative impact on the efficiency of the *approval* process, especially given the limited availability of expert safety assurance resources. However, this barrier needs to be overcome in order to realise the significant operational, financial and safety benefits which are available and which outweigh the increased cost of safety assurance. In addition, there was consensus within the ASCOS analysis that cost and efficiency of the assurance will improve in the medium and longer term as the ASCOS Method becomes established within the community.

Although the ASCOS Method has been refined taking feedback from the case studies and validation exercises, it has not yet been used in any actual applications within the industry. The ASCOS Method is presented here as

[1] Section 4 explains what is meant by a change in this context and presents several different ways of considering changes. The ASCOS Method can be applied to a wide variety of *changes*, not just safety enhancement systems or operations.
[2] Section 2.1 explains the distinctions made in this document between *approval* and *certification*.

an initial version, ready for application on real systems, but with the expectation that further improvements can be made in the light of experience.

1.4 Structure of this Document

The core of this document is the guidance on application of the ASCOS Method, which is presented in section 6. This section explains how to use the ASCOS Method to determine and follow an *approval path* for a change, supported by a modular *safety argument*. It also discusses how to get this *safety argument* agreed between the stakeholders and then to gather and generate the evidence which is presented together with the *safety argument* to the *approver* in order to gain *approval* for the change to be placed into operational service.

However, before presenting the ASCOS Method in detail, we present material which explains the ideas and techniques which underpin the method: it is important to understand these ideas in order to apply the method effectively.

Section 2 introduces and describes key concepts which underpin the discussion in the rest of this document; it is important that the reader takes time to understand these concepts in order to fully appreciate the presentation throughout the rest of the document.

In section 3 we provide a high level explanation of the ASCOS Method, showing the progress from initial identification of the need for a *change*, through development and deployment to monitoring in operational service.

We next explain (in section 4) what is meant by a *change* to the *TAS*, and cover the features of the *change* which need to be defined and considered at the very start of the application of the ASCOS Method.

This understanding of *change* provides the context for section 5, which introduces the concept of a *safety argument*. We explain how this concept can be used to support applications for *approval* of changes to the TAS and we present the generic *safety argument* which forms the framework of the ASCOS Method. We also introduce some tools for partitioning and managing the argument, especially when it becomes complex[3], and consider some of the pitfalls and problems with *safety arguments* which need to be avoided.

Once all the underpinning concepts have been established, section 6 presents the ASCOS Method in detail. This starts with identification and definition of the change followed by definition of the *approval path* which will be followed. This is captured as an *approval plan* which is presented to the *approvers* for agreement. The ASCOS Method then continues by following the development lifecycle of the *change*, developing the *modules* of the *safety argument* through the lifecycle and generating the evidence required to support the *safety argument*. Once the development and evidence is complete, the *modules* of the *safety argument* and supporting evidence are presented to the *approver* in order to gain *approval* for the *change*, with the aim of introducing the *change* into operational service.

[3] This complexity is usually inherent in the size and complexity of the TAS and the details which need to be considered when making any change to it.

Roles and responsibilities involved in applying the ASCOS Method are discussed in section 7.

After presenting the ASCOS Method in detail, we present (in section 8) the conclusions we have reached and recommendations for further work to develop and support the ASCOS Method.

Further supporting material is presented in appendices to this document.

The supporting rationale for the ASCOS Method, explaining how it has been developed through the lifetime of the ASCOS programme, and how it has responded to the experience gained from the Case Studies and the Validation Exercises, is presented separately from this document, in the final report for ASCOS WP1 (D1.6 [4]).

1.5 Typographic Convention

In the body of this document we have used *italic text* for terms with specific meanings: the meanings of these terms are defined in Appendix A.

2 Key Concepts

This section introduces several key concepts and explains how they have been applied within the ASCOS Method.

2.1 *Certification* and *Approval*

The term *certification* is widely used in the aviation industry. It describes the process of demonstrating that a physical item, or an organisation, meets a defined set of requirements and can therefore be issued a certificate to confirm this compliance. Certificates issued include:

- Type Certificates (TC) confirming that a particular aircraft type complies with the relevant certification basis
- Certificates of Airworthiness (CofA) confirming that a particular aircraft conforms to the type design (as defined by the Type Certificate) and is maintained accordingly
- Air Operator Certificates (AOC) confirming that an air operator complies with requirements set out by the national aviation authority for the operation of aircraft for commercial purposes
- ANSP certificates confirming that ANSPs comply with the common requirements for provision of ATM/ANS services

In other areas of aviation, alternative approaches are used, with *approval* being granted on the basis of a safety case or other document, without explicit issue of a certificate. For example, no certificate is issued either for *changes* to air traffic service provision by an individual ANSP nor even for products used to support the provision of air traffic services (ATS). However, these products and services are subject to *approval* by an *approver*.

Note: an advantage of *certification* over *approval* is that a certificate provides a confirmation of compliance which can be more readily reused in further applications of the certified product (or other entity).

The ASCOS Method is applicable to any *change* introduced to the *Total Aviation System (TAS)*. This means that the ASCOS Method is not restricted to *certification*. We have therefore deliberately avoided using terms which have a specific meaning within *certification* to avoid confusion when applying the ASCOS Method to *approval*. Where appropriate, we have separately mentioned the equivalent *certification* term to clarify the intention of the method. Note that the ASCOS Method can be used to obtain *certification*, and this would be especially relevant where the subject of *certification* goes beyond the existing standards in some way.

2.2 The Total Aviation System

Aviation must be considered as an integrated system where the elements interact in complex ways in order to deliver services including the transport of people and goods from one place to another. The overall system is referred to as the Total Aviation System (TAS) and includes concepts, equipment, people and processes.

Because of the complexity of the *Total Aviation System*, it is necessary to subdivide it in order to make any reasoning or *safety argument* manageable. The *TAS* can be subdivided into a number of *domains*, allowing each *domain* to be considered as a separate *module* of the *safety argument*, with *assurance contracts* established to record and manage the interfaces between the *domains* and with the external environment.

Appendix B provides a further description of the *TAS*.

2.3 Safety Argument

A *safety argument* is a connected series of statements, with supporting evidence, used to persuade the reader of the correctness of an overall claim or conclusion. It is not an argument in the sense of a disagreement.

Every time an *applicant* makes a request for *approval*, this is based on a *safety argument* of some form.

In many cases the *safety argument* is implicit in the procedures followed to gain *approval*; in other cases an explicit *safety argument* is presented in the *approval* submissions, e.g. by constructing a safety case. In some domains the *safety argument* can consist of both explicit and implicit components, for example the explicit requirements in a Certification Specification are often underpinned by implicit assumptions or context used in deriving those requirements.

The ASCOS Method is based on making an explicit *safety argument* to demonstrate to the *approver* (usually the relevant authority) that a particular *change* to the *TAS* achieves an *acceptable level of safety*.

The *safety argument* is split into *modules* aligned to the *domains* of the *TAS*. Dependencies between *modules* are captured in *assurance contracts*. This subdivision facilitates development of the *safety argument* separately in the individual domains, using approaches familiar within the domain, while the *assurance contracts* support the activity of ensuring that the *safety argument* remains consistent across the *TAS*.

As discussed earlier, the ASCOS Method covers a wider remit than just certification. As a comparison, the *safety argument* is in some ways analogous to the certification basis agreed with the *approver* at an early stage within the certification process. The *safety argument* is developed initially as a basis for agreement between *applicant* and *approver* on how *approval* for a change will be achieved. Evidence to support the *safety argument* is then gathered and / or produced during the development of the *change*: this is analogous to the certification evidence generated as required to satisfy the certification basis.

The concept of *safety argument* is further explained in section 5.

2.4 Making a change to the system

The ASCOS Method applies to *changes* made to the *TAS*; a *change* is any alteration to the *TAS*, beyond intended operational use or maintenance.

Thus *changes* range in scope from upgrade of existing equipment items all the way through to introduction of a new operational concept. The ASCOS Method can be applied to any of these *changes*.

It is important to recognise that a *change* may have wide ranging impact across the *TAS*, beyond the immediate part of the *TAS* which is being changed. A key part of the ASCOS Method is to perform a complete evaluation of the safety impact of the *change* in order to support the overall *claim* that the *change* achieves the agreed *acceptable level of safety*. The success of this evaluation depends on a thorough understanding of the *TAS* and the interactions between the parts of the system.

The important concepts relating to changes are further explored in section 4.

2.5 Acceptable Level of Safety

The ASCOS Method focuses on demonstrating that the *change* delivers and continues to deliver an *acceptable level of safety* across the *TAS*. In other words, the level of safety after the change must be acceptable to all competent authorities who are affected by the change. Note: this does not necessarily mean that an improvement in safety must be demonstrated, but there is a general desire to seek ways to reduce risks as far as practical. A *change* may be necessary to address specific risk escalations but generally it is more likely that the purpose of a *change* is to improve operational capability

It is therefore necessary to determine appropriate safety criteria in each *domain* affected by the *change* and separately demonstrate that these are met in each case. Such criteria may be either absolute (specific safety objectives and integrity requirements based on apportionment of a safety target) or relative (comparison of the risk prior to the change against the predicted risk following the change, on the premise that the prior risk is tolerable). In the civil aircraft domain, the existence of the target for a catastrophic failure of 10^-9 per flight hour makes it much easier to apportion absolute targets, whereas the absence of (and difficulty of defining and agreeing) similar absolute targets in other *domains* means that relative targets are often used.

A *change* which decreases safety (i.e. increases safety risk) in one *domain* is usually difficult or impractical to justify, even if it significantly increases safety in other *domains*[4]. To trade off safety between *domains*, it would be necessary to provide a robust quantification across all *domains* which demonstrates a significant overall positive impact on safety. Production of such a robust quantification is made more difficult by the fact that different *domains* use different types of targets (often with different units), making it difficult to create valid comparisons between *domains*. A corresponding assessment would be needed in the event of a *change* with differing impacts on different sovereign states. (A recommendation for further research in this area is made in section 8.3.7.)

In the past, there has been a tendency to consider only the risks resulting from failure of the new system, giving rise to an unnecessarily negative assessment of the impact of the *change*. Any evaluation of safety must take into account both

- any improvement in safety intended by the change and
- any risks introduced by the change

[4] It is self-evident that a *change* which decreases the overall safety of the *TAS* will not be acceptable.

Further discussion of the concept of acceptable safety is presented in section 6.3.8.

2.6 Criticality of interfaces

Many changes will involve multiple organisations and affect multiple parts of the *TAS*. Achievement of safety is critically dependent on all these parts interacting correctly with each other and with the environment. However, there is a significant risk of misunderstanding (and therefore incorrect interaction) between different parts of the *TAS*. This risk is often exacerbated by the differing perspectives and priorities of the different organisations responsible for the parts of the *change*.

It is therefore critical to ensure that the interfaces between different parts of the system are fully defined in a way which is understood and accepted by (all) stakeholders affected by the interface.

The ASCOS Method supports this with the concepts of:

- modularisation, where the *safety argument* is subdivided into *modules* aligned to the subdivisions within the system
- *assurance contracts*, where the dependencies between the *modules* are expressed formally and managed as part of the *safety argument*.

The concepts of modularisation and assurance contracts are discussed further in section 5.3 and section 6.4.

2.7 Performance based vs compliance based approaches

Approaches to *approval* are often characterised as either performance based or compliance based.

This terminology can be used to distinguish between:

- requirements or targets which are relatively high level and solution independent (performance based)
- requirements which are expressed as a detailed set of constraints often assuming a particular solution

The terminology can also be used to distinguish between:

- the goal based approach often used in ATM
- the certification based approach often used in the aircraft domain

Although there is overlap between these two different ways of viewing the approaches, it is useful to bear both views in mind.

One concern driving the ASCOS Project is that parts of the aviation industry have historically taken a compliance based approach to *approval* and that this approach stifles innovation because specifications based on historical solutions can be difficult to apply to novel solutions. The performance based approach has been suggested as a way of allowing developers the freedom to innovate and therefore develop optimal solutions.

In practice most approvals use a mixture of approaches – for example, CS25.1309 [8] is goal based whereas a large proportion of this CS is based on compliance with specific prescriptive requirements.

The ASCOS Method allows a goal based approach using high level, solution independent targets to support the development and assessment of innovative solutions, while also allowing more detailed requirements to be used to ensure consistent application of established solutions. Prescriptive requirements (a compliance based approach) are also useful to constrain interfaces or express well established rules, especially where these relate to interfaces with parts of the *TAS* unaffected by a *change*.

2.8 Keeping existing processes where relevant

The existing processes in the aviation industry have served very well to achieve and maintain high levels of safety over many years. These processes and standards remain relevant and they are not replaced by the ASCOS Method.

However, the aviation industry is now facing an increasing number of changes which:

- introduce innovative technologies and concepts – challenging compliance-based processes
- affect multiple parts of the *TAS* – necessitating the application of different sets of processes

The ASCOS Method provides a framework for evaluating the existing processes and then adapting or augmenting them to face these challenges, while retaining the good experience captured by the existing processes where this remains relevant and applicable.

The concept of evaluating and adapting or augmenting existing processes as necessary is covered further in section 6.4.

2.9 Consistent Terminology

Clear communication depends on clear and consistent definitions of the terms used. This presents a particular problem for the ASCOS Method, because some terms have different meanings in different domains.

In this document, a number of specific terms have been used to describe the ASCOS Method and associated concepts: these terms are listed and defined in Appendix A. Where these terms are used in this document, they are shown in *italic* type.

Where terms already have an accepted and consistent meaning within the industry, they are used with the same meaning within the ASCOS Method.

Where it was necessary to express a meaning different from the accepted meaning within the industry, a new term has been introduced rather than adapt / alter the meaning of an existing term.

Where the meaning of a term is inconsistent within the industry this is highlighted.

3 The ASCOS Method

The ASCOS Method provides a framework for obtaining safety[5] *approval* for any *change* to the *TAS*. However, the wide variety of potential *changes* means that it is not possible to provide a detailed step-by-step description of the process for obtaining *approval*. Instead, the method should be seen as a framework providing guidance on how to obtain *approval* for any change to the TAS. The method recognises that the process to be followed will depend upon the type and scope of change being made.

The ASCOS Method can be applied to *changes* which include an element of *certification* (e.g. granting of a certificate of airworthiness) and can be used to develop the *certification basis* and certification plan for such *changes*. However, the method is not limited to such *changes*: it can also be used where no certificate will be granted, for example where a new operational concept is being introduced into operational service for the first time. Such *changes* need planning and *approval*, but may not be subject to *certification*.

The framework is constructed around developing an *approval path* and supporting *safety argument* for a change. This section explains (at a high level) how to scope, develop and refine the high level *safety argument* in order to gain *approval*. The concept of *safety argument* is discussed in detail in section 5. The detailed description of what should be considered at each stage is provided in section 6.

An overview of the overall ASCOS Method is presented in section 3.1. Next, the concept of an *approval path* is introduced in section 3.2 and the application of the ASCOS Method is explained further in section 3.3.

Note: roles and responsibilities for the various steps of the process are discussed in section 7.

3.1 Overall View

The ASCOS Method can be viewed as a process starting with the first identification of the need for *change*, all the way through to the monitoring of the *change* in operational service, as illustrated in Figure 2.

[5] The ASCOS Method has been developed to specifically consider gaining *safety* approval. It could be used to address non-safety requirements, but the greatest benefits of the approach are achieved where a detailed assurance argument needs to be built spanning multiple *domains*, which is typically what is needed to demonstrate that safety requirements are met.

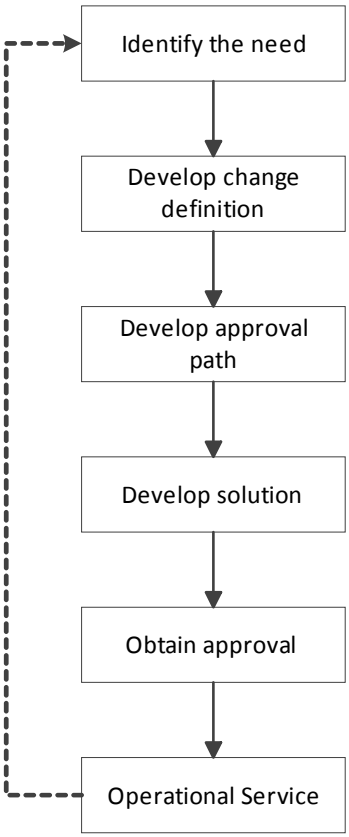

Figure 2: Overall View of ASCOS Method

The steps are as follows:

1. **Identify the need**: The needs for *change* to the *TAS* can be understood in the following broad groups:
    a. business need
    b. a specific need to improve safety, in response to monitoring current performance
    c. external changes

    In each case, the first step is to identify the (potential) *change* and identify the *change leader*.

2. **Develop change definition**: Before deciding how to gain *approval* for the change, and who needs to be involved, the change must be defined sufficiently to understand:
    a. what is being changed
    b. who is responsible for making the *change* (this may include multiple organisations, but should usually be led by a single organisation or individual – the *change leader*)

c. the conceptual solution[6]

d. who/what is affected by the *change* (this should include everyone affected, including effects which may not initially be apparent)

e. what regulations apply to the change

f. factors in the environment[7] which constrain the *change*

g. the *acceptable level of safety* for the change

h. who is/are responsible for giving *approval* for the change to enter operational service

In practice, definition of the *change* continues throughout the process; however it is critical to have a well-defined baseline definition of the *change*, including its environment, at the outset, allowing any later variations to be properly evaluated and incorporated. The definition and evaluation of *changes* is discussed further in section 4.

3. **Develop approval path**: The *approval* path depends on the details of the *change*. For some *changes*, it will be sufficient to follow existing *approval* approaches with little or no variation; for other changes, a new *approval* path must be defined because none currently exists, or because the current path is too costly (either in resource or time). The *approval* path is supported by a *safety argument* which justifies the claim that the *change* will meet the *acceptable level of safety*. The *safety argument* is divided into *modules* aligned to the *domains* of the *TAS*. The development of an *approval path* is discussed further in section 3.2. The *approval* path is documented in an *approval plan* and agreed between *applicant*(s) and the relevant *approver(s)* before development commences.

4. **Develop solution**: The next step is to develop the solution (i.e. how the change defined in step 2 will be implemented) and gather the evidence needed to support application for *approval*. Development occurs at two main levels: (a) at the level of the *TAS*; and (b) within the individual *domains*. This development is iterative until the development is complete and is explained further in section 3.3.

5. **Obtain approval**: At completion of development, the *modules* of the *safety argument* (together with supporting evidence) are submitted for *approval* in accordance with the plan which was previously agreed with the *approver(s)* involved. (The *approver(s)* will grant *approval* (only) when they are convinced that the *safety argument* and supporting evidence demonstrate that the *acceptable level of safety* has been achieved.)

6. **Operational service**: Once *approval* has been gained, the *applicant* informs the relevant stakeholders of the details and timescales of the change, and then brings the change into operational service. Following entry into service, the operation is then monitored in accordance with the relevant organisation(s)' SMS to confirm that the *acceptable level of safety* is achieved and maintained. Where the level achieved is not acceptable, the *change* may be withdrawn from operational service; otherwise further changes are designed and implemented to rectify the deficiency.

[6] Although the detailed development of the solution takes place later, the concept solution must be defined in order to identify which regulations will apply and which parts of the system will be affected.
[7] Environment is here taken to include operational and regulatory environment as well as the physical environment.

Although the ASCOS Method is presented as a linear path, iteration is required for most changes. This could arise either because the requirements change, or because development has revealed that the original approach needs to be adapted. It is also relevant to note that where the change affects the *approval* path, the *approval plan* should be revised and re-presented to the *approver* at the earliest opportunity to confirm that the proposed *approval* path remains acceptable. If this is not done, the *applicant* risks taking a path which will not be acceptable to the *approver*, leading to rework late in the process, which may be both expensive and time consuming.

3.2 Approval Path View

The overall intention of the ASCOS Method is to gain *approval* for a change to the Total Aviation System (TAS). *Approval* is granted by the *approver* on the basis of a *safety argument* (supported by evidence) justifying that the change will be acceptably safe. Although the concept of *safety argument* may be unfamiliar, it is already implicit in current approaches to gaining *approval* (or certification). The concept of *safety argument* is explored in more detail in section 5. (An illustration of how elements of a *safety argument* can be implicit is given in section 5.1.3.)

The ASCOS Method can be viewed as establishing an *approval path* which, where possible, is based on existing approaches (which provide the evidence required by the current, often implicit, *safety arguments*).

For some *changes*, the *approval* path can be based entirely on existing approaches and appealing to the existing (possibly implicit) *safety argument*. This is a valid approach where:

1. the existing approaches are fully applicable to the *change* being made;
2. the existing approaches fully consider all the impacts of the *change*; and
3. there is no (safety or efficiency) benefit to be gained from improving the approach.

An example of such a *change* might be the introduction of an upgraded equipment item on board an aircraft, where the new item has the same fit, form and function as the existing item. This could be visualised as a straight, already-established path, as shown in Figure 3.

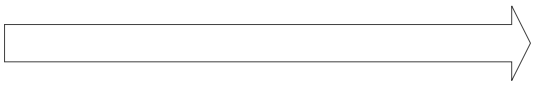

Figure 3: Approval path using existing approaches

For other *changes*, established approaches will provide the majority of the evidence needed, but with some gaps. For example, the *change* may introduce a novel solution which is not covered by the existing approaches, as shown in Figure 4.

Figure 4: Novel solution not fully covered by existing approaches

In this case, the *approval path* may be established by developing approaches which cover the novel solution. These approaches must be developed in a way which takes account of the interface between the novel parts of the solution and the rest of the solution, to make sure that these are fully considered and integrated. The development of these additional approaches provides the missing part of the *approval path* for the solution. This must then be supported by a *safety argument* which demonstrates that the combination of existing and new approaches fully addresses the change and that the resultant solution achieves the *acceptable level of safety*.

Figure 5: New approaches developed to complete the approval path

In some cases, the existing approaches may be sufficient to provide an *approval* path, but a more efficient (and therefore cheaper) approach may be possible. The development of additional approaches improves efficiency, as illustrated in Figure 6. As before, these new approaches must be supported by a *safety argument* which demonstrates that the combination of existing and new approaches fully addresses the *change* and that the resultant solution achieves the *acceptable level of safety*.

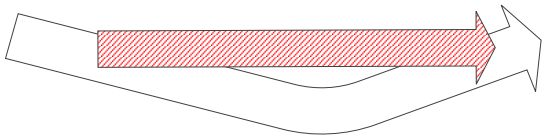

Figure 6: New approaches developed to provide more efficient approval path

In other cases, there may not be any existing approaches, and the *approval* may need to be developed entirely from first principles, as illustrated in Figure 7.

Figure 7: Development of entirely new approval path

Complex or large changes may involve a combination of the above, such that some parts may be approved straightforwardly whereas others may require additional approaches to be developed and still others may allow for a more efficient approach, as illustrated in Figure 8. Note that in these cases it is important to review the approaches against each other to ensure that the overall approach remains consistent in achieving the overall objective of a safe *change* to the *TAS*.

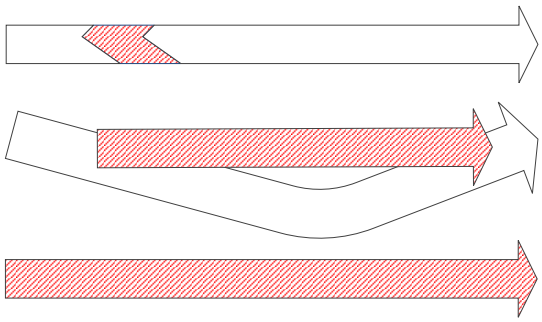

Figure 8: Different approval paths for different parts of the system

In each case (with the possible exception of where the path exactly follows the existing approaches), a *safety argument* is needed to demonstrate that the change achieves the *acceptable level of safety*. However, the scope of the *safety argument* required depends on the degree of novelty involved and on the degree to which the *change* spans multiple *domains* of the TAS. (*Safety arguments* are addressed in more detail in section 5.)

Note that development of *changes* is a complex process. It is rare (or even unknown) for the full definition, impact and scope of a *change* to be understood at the outset. The *approval path* should be re-evaluated regularly to check whether the remaining *approval path* is (a) complete and (b) efficient.

3.3 Development View

The detailed development of the solution and *safety argument* proceeds at two levels, and in a cyclic manner, as illustrated in Figure 9. Development and evaluation of both solution and *safety argument* occurs in parallel in the steps shown, as further described below. The arrows show progression through steps of the process.

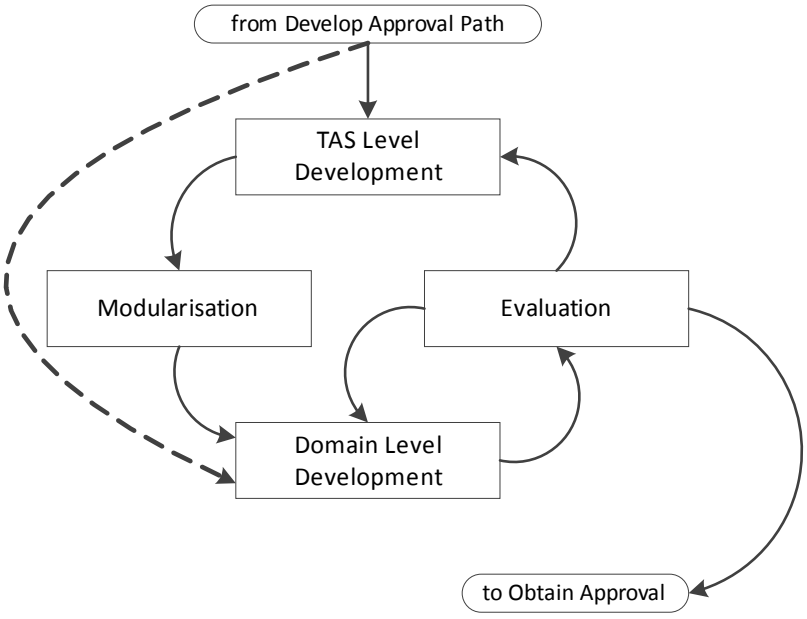

Figure 9: Cyclic development of solution and safety argument

The initial development is at the *TAS* level. For some wide ranging changes, this will involve significant systems engineering and analysis to define and assess the *change* at this level. Even for *changes* where only a small adaptation of the existing approach is needed, a review at the *TAS* level is needed to ensure that the overall impact on the *TAS* has been fully considered. At this stage the *safety argument* is also developed at the *TAS* level.

Development at the TAS level is followed by modularisation of the change into subparts aligned to individual domains. This modularisation includes defining the requirements to be satisfied by the individual subparts, as well as defining the *assurance contracts*[8] between the subparts.

This modularisation is followed by development within individual *domains*. This domain level development may proceed according to the existing approaches within the *domain*, depending on the choice of *approval path* (see section 3.2). Alternatively, new approaches and arguments may be needed within the *domain*. These may be needed to support the development of novel solutions; they may also be needed to ensure that the interfaces with other domains are fully addressed within the development.

For changes where the major impact is within a single *domain*, the detailed development may be limited to that one *domain*, supported by establishing the impact which the *change* has on other *domains* and ensuring that this is fully captured in *assurance contracts*.

During development, evaluation is also necessary to check that the solution and *safety argument* remain consistent and complete at a number of levels:

- Does the solution at the *TAS* level still meet the requirements of its stakeholders?
- Does the solution being developed in the individual *domains* meet the requirements imposed at the *TAS* level?
- Does the evidence being produced continue to support the *safety argument*?
- Does the solution satisfy the *assurance contracts* between the *domains*?

(From an *approval* perspective, the *approver* will be interested only where the answers to these questions have an impact on the *approval* – i.e. on the *safety argument* and / or the supporting evidence. In this context the *approver* is one of the stakeholders in the development of the *change*.)

If the answer to any of these questions is “No”, then it is necessary to go back and revise the solution and / or the *safety argument* to ensure that the overall development remains on course.

Large programmes are often divided into a number of lifecycle stages, with “stage gates” between stages. (E.g. concept design, detailed design, realisation, test and deployment.) The programme must be able to demonstrate that certain criteria are met before it can proceed to the next stage of the lifecycle. The stage

[8] An *assurance contract* defines the assurance which parts of the system need to provide to each other in order to satisfy the overall *safety argument*.

gates may be an appropriate point at which to evaluate the state of the development and the *safety argument* and to take corrective action as necessary.

System development, especially on large programmes, is often subject to variation during the lifecycle, and this variation can come from any number of sources. Examples of this include:

- introduction of a new aircraft type, leading to the need to accommodate this aircraft in the design of the aerodrome
- discovery of an incorrect assumption made during concept development (e.g. wake separation requirements) leading to alteration of the concept

It is recognition and successful management of these variations which is the biggest challenge in the development of *changes*; it is here where the roles of the *change leader* and *argument architect* become crucial to ensure that the *change* continues to meet its requirements and that the *safety argument* continues to support the *change*.

This development and iteration continues until the development is complete and the *safety argument* is fully developed, consistent with the development and supported by evidence.

4 Understanding and Handling *Change*

The ASCOS Method is applied to obtain *approval* for a *change* to (a component) of the *TAS*. Before considering the ASCOS Method, it is important to understand the *change* itself.

This section explains what is meant by a *change* and looks at the relevant features of the *change* which need to be considered before deciding how (and whether) to apply the ASCOS Method – in particular it is important to understand the impact of the *change* as fully as possible. This section also looks at the lifecycle of a *change* and its effect on the application of the ASCOS Method.

If this section is read with a specific *change* in mind, it should allow the reader:

- to decide whether the *change* is one for which the ASCOS Method should be considered
- to form an initial view of the scale of effort required to apply the ASCOS Method
- to make a preliminary assessment of the impact of the *change*
- to identify the stages involved in the *change*

4.1 What is a *change*?

A *change* is any alteration to the TAS, beyond intended operational use or maintenance. Such *changes* need some form of *approval* before they are implemented. Usually the *approval* will be given by a competent authority (e.g. EASA or the relevant national authority) but the ASCOS Method could also be used where the organisation making the *change* has the authority to grant its own *approval* – this is why the term *approver* is used to encompass the wider concept.

This definition encompasses a wide range of *changes*, including:

a. introduction of new or replacement equipment items

b. introduction of a new concept, such as self-assured separation

c. changes of airspace structure (e.g. new routes, or change in transition altitudes)

d. granting permission to a new organisation (e.g. air operator) to operate (or varying their scope of operation)

e. changes to regulations or standards

This list is not exhaustive – the ASCOS Method can be applied to any *change* which needs *approval* before it enters service. (However, it should be noted that the degree of effort required will vary significantly with the type of *change*.)

It is important to note that for any *change* to the *Total Aviation System*, a *safety argument* is needed. In particular, where a trial operation is introduced (perhaps as part of a proof of concept), it is still necessary to demonstrate that the trial achieves the relevant *acceptable level of safety*[9]. Where equipment or procedures are not fully proven, this *safety argument* may be based on mitigations which are in place to limit the scope for harm in the event of failure – for example flying in segregated airspace, or flying without passengers on board.

4.2 Broad Types of *Change*

Changes can be placed into the following types:

1. replacement of equipment item with a similar item, with form, fit and function unchanged
2. *change* within a single domain, although it may have an impact on other domains
3. *change* across multiple domains

Examples of types 2 and 3 are given in section 4.3.

In practice, a change may not fall neatly into a single category: the framework described in this document should be used as a guide and adapted as appropriate to the specific *change*.

The level of effort needed to apply the ASCOS Method will usually vary depending on the type of *change*. For type 1 *changes*, it will usually be possible to gain *approval* through applying existing approaches (for instance by showing compliance with the corresponding ETSO[10]). For type 2 *changes*, more effort will be required to develop the *safety argument* for the *change*, although the approaches normally used within the *domain* are likely to form a significant part of the *safety argument*. Where a significant amount of novelty is involved, it is likely that significant new approaches and standards will need to be developed to cater for this novelty. Type 3 *changes* are likely to involve the largest amount of effort to understand the impact of the *change*, including the interactions between *domains*, and to develop the evidence to demonstrate the safety of the change accordingly. Again, the development of new approaches and standards is likely to be required.

4.3 Impact of *change*

When a *change* is made to some parts of the *TAS*, there may also be effects on unchanged parts of the *TAS*. Some of these effects will be intended (e.g. introduction of an autopilot reduces the flight crew's workload); in addition there are often unintended effects (e.g. the increase of automation on the flight deck reduces the crew's level of familiarity with some operations). The collection of these intended and unintended effects is termed the impact of the *change*.

[9] In other words, the trial does not have an unacceptable negative impact on the safety of the (operational) *TAS*.
[10] An European Technical Standard Order (ETSO) is a way to demonstrate that a part complies with a minimum performance standard.

Some *changes* have minimal impact on the rest of the *TAS* (e.g. introduction of a replacement equipment item which has the same form, fit and function as the existing item); other *changes* have a much wider impact (e.g. introduction of a new aircraft type).

This variety of impact is roughly illustrated in the following figures. In each case, the type of *change* (from the list in section 4.2) is also indicated.

Note: these figures are roughly aligned to the EASA Regulations Structure (see Appendix B), although this has been simplified for the purpose of the illustration here.

- Figure 10 depicts a much simplified picture of the *TAS.*
- Figure 11 (type 1) depicts the potential areas of the *TAS* which would be affected by a simple *change* to the non-co-operative surveillance systems at an airport.
- Figure 12 (type 2) depicts the potential areas of the *TAS* which would be affected by an Automated Aircraft Recovery System (AARS)[11] which could be engaged by the pilot to return the aircraft to stable flight to allow the pilot opportunity to regain situational awareness.
- Figure 13 (type 3) depicts the potential areas of the *TAS* which would be affected by introduction of Remotely Piloted Aircraft Systems (RPAS) into unsegregated operation in civil airspace.

[11] One of the ASCOS case studies considered such a development and illustrations from this are used at various points within this document.

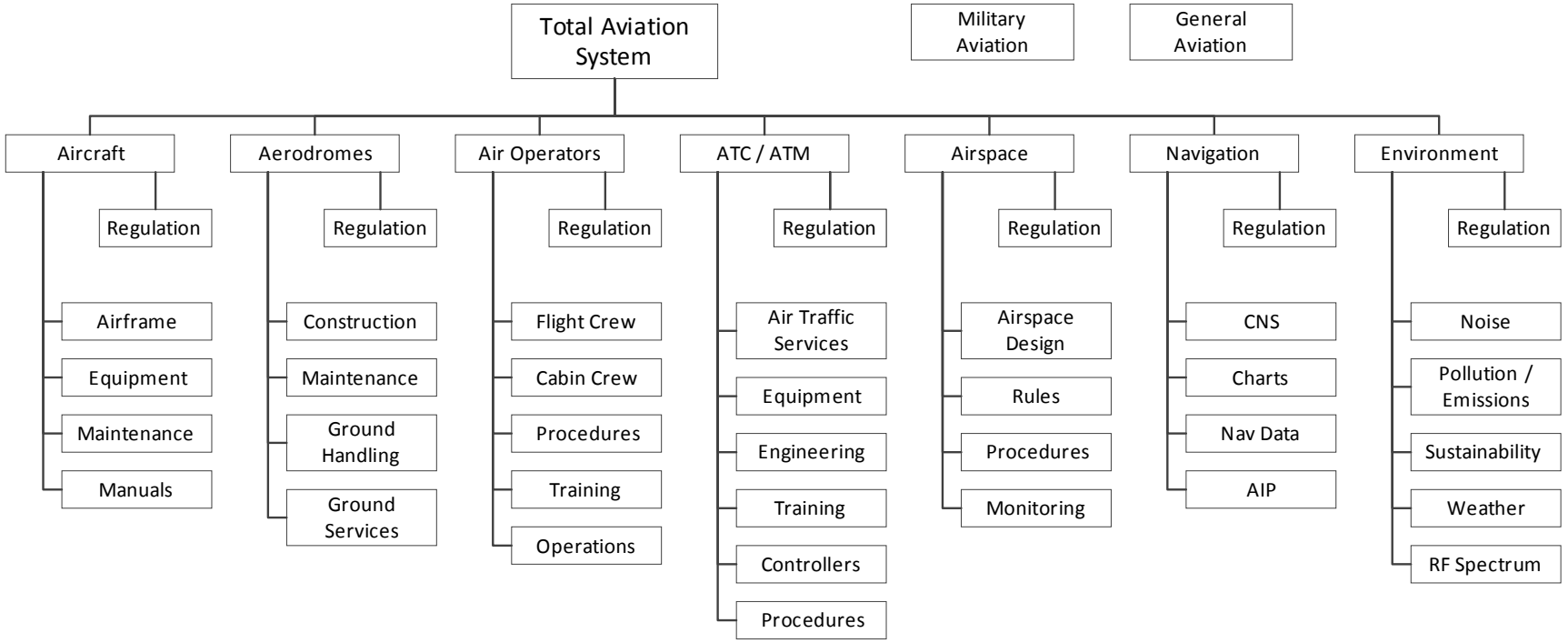

Figure 10: Illustration of breakdown of TAS

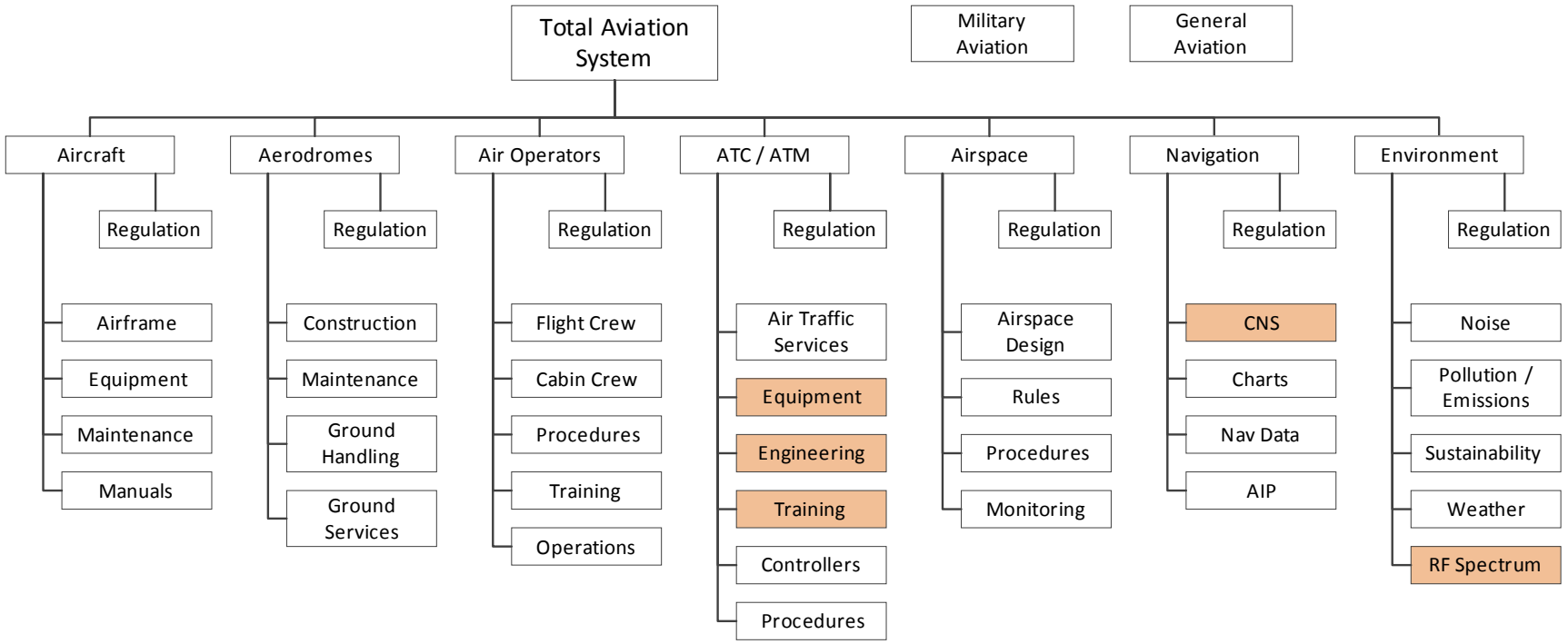

Figure 11: Impact of introduction of new non-co-operative surveillance system

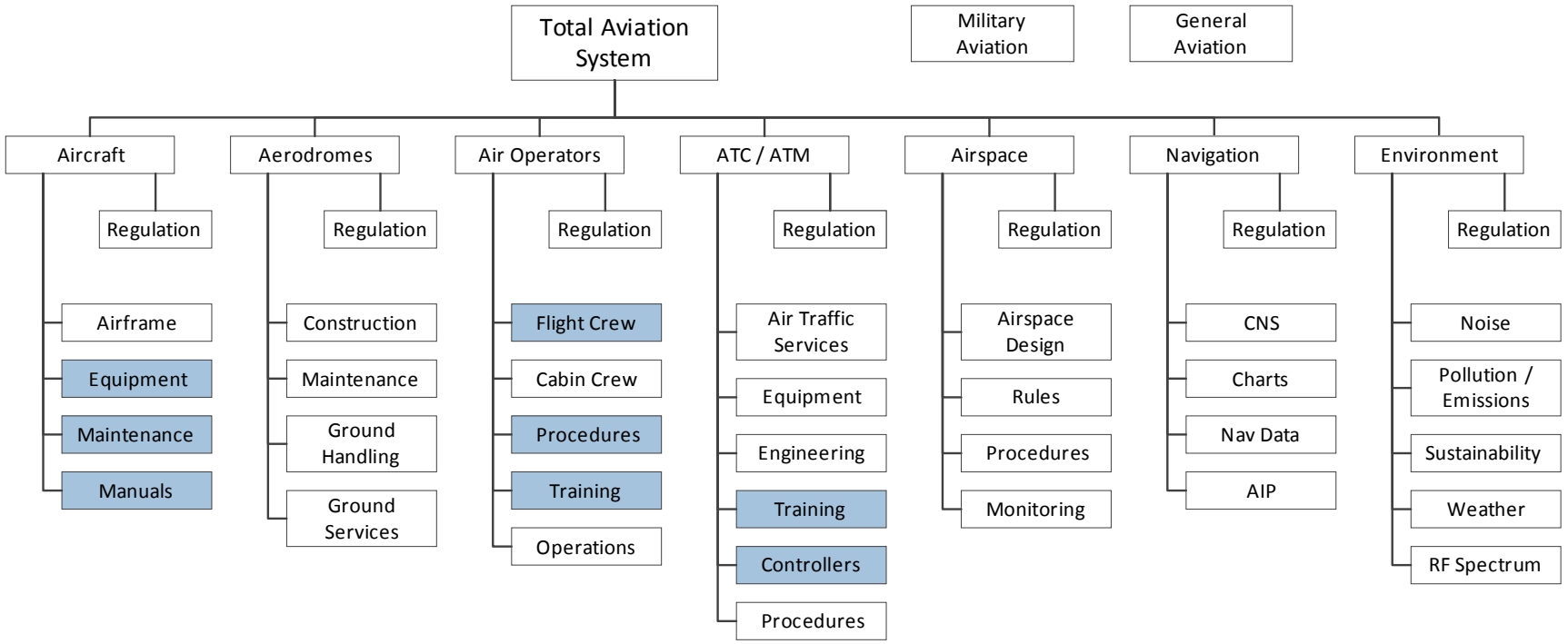

Figure 12: Impact of introduction of automated aircraft recovery system (AARS)

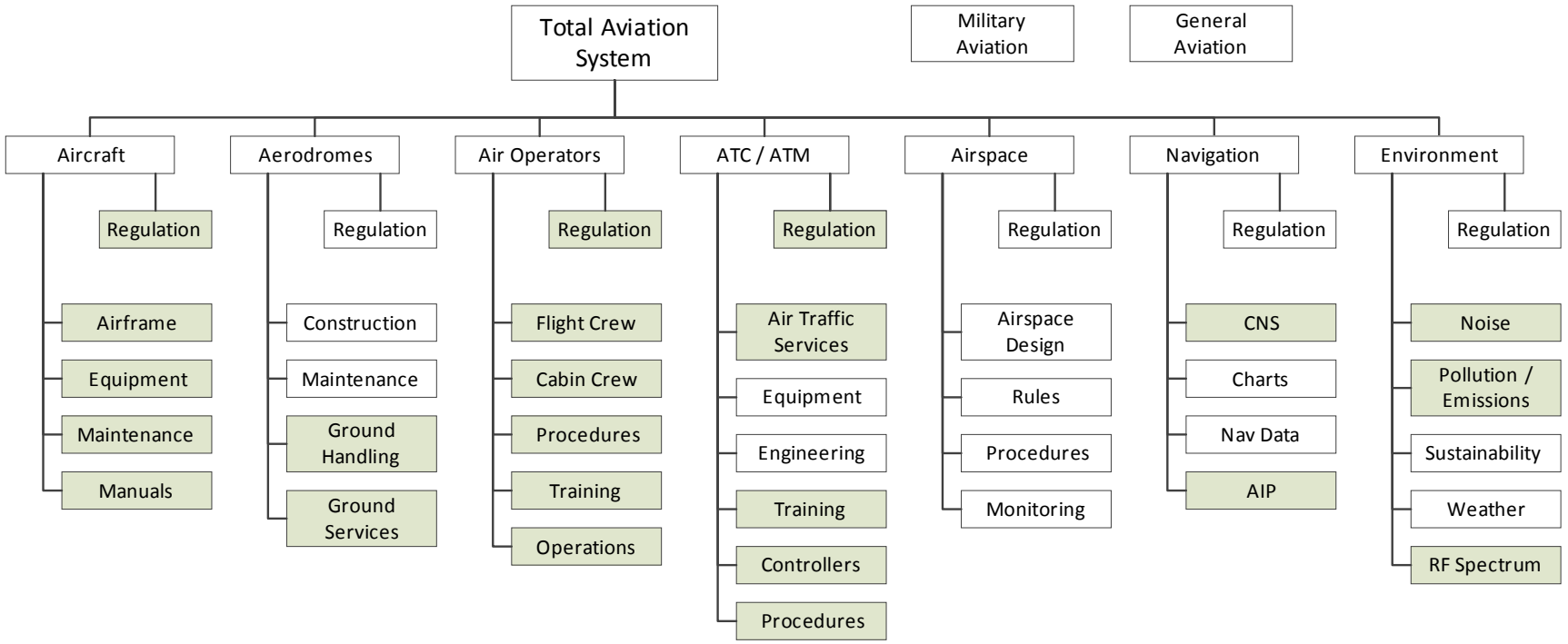

Figure 13: Impact of introduction of RPAS in non-segregated airspace

It is important to undertake an early evaluation of the impact of a *change*, in order to identify the stakeholders who need to be involved, the regulations which need to be complied with[12] and the *approvers* which will approve the *change*. As discussed below (see section 4.4), the understanding of the *change* develops through the lifecycle. The initial impact assessment will necessarily be at a high level, with detail added as the understanding of the *change* develops.

Impact analysis must consider the whole *TAS* and all service configurations and modes of operation, including those associated with fallback or degraded modes. It must consider any intended safety improvement due to the change as well any safety effects due to failure of the introduced systems. Explicit consideration of the safety benefits arising from the *change* is a crucial step which can be overlooked in a purely failure-based approach. The need for this part of the *safety argument* (sometimes termed the "success approach" [9]) arises because there are *inherent hazards* within the *TAS* (e.g. conflict between aircraft trajectories or controlled flight towards terrain) which systems such as ATM systems are introduced to prevent. It is necessary to confirm that, where a *change* is intended to deliver a safety benefit (i.e. reduce the risk of *inherent hazards*), it does indeed deliver a sufficient safety benefit. Too much focus on the analysis of failures within the *TAS* may divert attention from the crucial question of whether the design (when functioning correctly) provides sufficient mitigation of these *inherent hazards*.

A thorough impact analysis is necessary to ensure that all possible effects on safety, whether intended or unintended, are identified and evaluated. If the analysis is incomplete, areas affected by the *change* may be missed, which may result in an unacceptable (and not initially detected) degradation of safety.

Table 1 lists areas which should be included in the impact analysis along with some examples from the case study examining introduction of an RPAS. Note: this table should be used only as a guide, and is not intended to constrain the impact analysis.

Area to consider	Examples (mostly from introduction of an RPAS)
Intended effects of the *change* throughout the *TAS* (including operational and maintenance changes)	Need to consider effect on ATM, maintenance, crew licensing, aerodrome requirements, not just impact on airworthiness requirements
Unintended effects of the *change*	Lack of availability of spares through conversion of old aircraft to RPAS rather than decommissioning
Changes in failure scenarios (including new / changed failure modes and failure rates, potentially new common cause failures)	New communications links whose failure will have a significant effect on system safety
Changes in the effect of failures (including changes to failure detection and correction, failure propagation, responses to failures)	Removal of personnel from flight deck removes failure detection by human senses (e.g. smell, temperature sensation, visual of weather conditions); alternative means of detection are therefore needed

[12] In a *certification* this would be equivalent to agreeing the *certification basis*.

Area to consider	Examples (mostly from introduction of an RPAS)
Effect at interfaces between parts of the system	Interface between pilot and control surfaces subject to increased processing and data transmission, introducing potential delays and failure modes
Effect on and of the environment[13]	Environmental impact of additional emergency landing areas for RPAS; RPAS systems will have different capabilities in respect of dealing with ranges of environmental conditions
Proximity of components or environmental elements (e.g. visual reference points)	*Not specifically from RPAS, but where new components are introduced, what physical (e.g. heating) effect can they have on the components around them.*
Operation outside critical thresholds (e.g. of resource usage)	De-skilling of pilots through increased automation
Changes to stress, capacity or loading	Introduction of additional communications channel (ground to ground comms between ATCO and (ground-based) pilot)
Positive or negative feedback effects	Aircraft to ATM communications have built in confirmation loops, which may be affected by communications delays where the pilot is remote from the aircraft

Table 1: Areas to be considered in impact analysis

Impact analysis is primarily a search for connections of any type between the changed parts and other parts of the *TAS*, either via a functional interface or a shared resource. These connected parts are then analysed to determine whether their behaviour is affected by the *change*. Analysis also determines whether the conditions experienced by the connected parts have changed, requiring an extension to those parts' specifications. (This may in turn lead to identification of additional *approval* / *certification* work for these affected parts.) When the behaviour of an impacted part is changed, then the impact analysis must extend further to identify whether further parts are affected in turn.

At this stage, a review of existing models of the affected parts of the TAS can assist in evaluating the impact of a change. This includes the models and tools developed by ASCOS for safety risk management (as described in the WP3 Final Report [5]). The models can be used as part of the "identify the need" step (see section 6.2) to identify (from the performance indicators) where improvements are needed. They can also be used to understand how a *change* in one area might propagate through the *TAS*. However, it is important to fully understand the scope of any model used: a model can only support evaluation of impact in areas which are covered by the model.

The *safety argument* for a *change* must include a complete evaluation of all aspects of the *change* which may affect safety in order to determine whether the *change* delivers the *acceptable level of safety*.

[13] Here "environment" is meant in its widest sense, including items relied on by the crew – such as visual reference points – as well as weather, pollution, noise, etc.

4.4 Progressive Understanding of *Change*

Changes follow a (system engineering) lifecycle from initial proposal through introduction into operation, culminating in monitoring of the *change* while in operation. *Approval* activities follow a different lifecycle, although a rough mapping can be made. This is illustrated in Table 2 which shows the mapping between the engineering lifecycle presented in the European Operational Concept Validation Methodology (E-OCVM) [6] and the *approval* lifecycle proposed by the ASCOS Method. It should be noted that *approval* activities tend to be focused at particular stages within the engineering lifecycle.

<table>
<tr><th>E-OCVM Lifecycle Stage</th><th>ASCOS Method Stage</th></tr>
<tr><td>V0 (System Needs)</td><td>Identify the need</td></tr>
<tr><td>V1 (Scope)</td><td rowspan="2">Develop change definition</td></tr>
<tr><td>V2 (Feasibility)</td></tr>
<tr><td rowspan="2">V3 (Preindustrial development and integration)</td><td rowspan="2">Develop approval path</td></tr>
<tr><td></td></tr>
<tr><td>V4 (Industrialisation)</td><td rowspan="2">Develop solution</td></tr>
<tr><td>No direct mapping</td><td>Obtain approval</td></tr>
<tr><td>V5 (Deployment)</td><td rowspan="2">Operational service</td></tr>
<tr><td>V6 (Operations)</td></tr>
</table>

Table 2: Mapping the ASCOS Method to the E-OCVM lifecycle

When a *change* is first proposed (V0, V1), only a limited amount of information is available. For example, a proposal to introduce a new surveillance system (for use by air traffic controllers) may initially be defined as a performance requirement for monitoring position and speed of aircraft, without any constraints on the technology used. The models and tools developed by ASCOS for continuous safety monitoring (as described in the WP2 Final Report [7]) and other similar tools can be used here to identify and evaluate the need for a change to the system. However, it is important to fully understand the scope of any model used: a model can only support evaluation of impact in areas which are covered by the model.

It is important to make an early assessment of the impact of a *change*. At this stage it may only be possible to identify the main areas affected by the change and the main impact on those areas. Evaluation and consultation with the *approver* will indicate the *approval path* to be taken by the change and the degree of *approval* effort required: a change with significant impact on multiple *domains* will need a more thorough *safety argument* than a *change* whose impact is limited to a single *domain*. However, even at this stage, the *change* may be found to be non-viable and the development may be stopped even before a start is made to building the *safety argument* required by the ASCOS Method.

As the *change* becomes further developed (V2), there will be a clearer understanding of the parties which will be affected and of the impact of the *change*. At this stage the *approval path* (see section 3) can be established along with the outline *safety argument* and *approval plan*. It is still possible that the negative impact on safety

is revealed to be too great, such that the *change* is judged to be not viable and the development may be stopped.

The *change* will then go through stages of detailed design and implementation (V3, V4, V5), with corresponding assessment activities in parallel. During these stages the *safety argument* may need to be updated, with knock-on effect on the *approval plan* (see section 6.4.5).

As discussed in section 3, it is quite likely that the definition of the *change* will itself be altered during the lifecycle of the *change*. It is critical that these alterations are properly managed, through a change management and impact assessment process, so that the development and assessment remain consistent with the definition of the change.

4.5 Staged *Changes*

Complex *changes* are often developed in multiple stages. There are two ways in which this can occur:

1. Stages aligned to different parts of the system lifecycle
2. Stages aligned to different operational states of the final system

In each case, it is important to understand the overall nature of the *change* and the impact which this will have on the application for *approval*, as illustrated in the following sections.

4.5.1 Stages reflecting different parts of the system lifecycle

Development of a new aircraft concept is a complex process with different parties involved at different stages. This can be considered in multiple stages as illustrated in Table 3, taking the example of the development and introduction of an RPAS operating in unsegregated airspace.

Stage	Description	Change Leader	Outputs
1	Development of requirements for an RPAS operating in unsegregated airspace; may be undertaken by a pan industry consortium of manufacturers and operators, all of whom are jointly interested in ensuring that the requirements meet their needs	*TAS Engineering and Safety Group (TESG)* – see section 7.1.5	Approved requirements
2	Implementation of the requirements to generate a specific RPAS product; the details of the development would be commercially confidential and would not be shared between manufacturers, and lead to a product Type Certificate	Manufacturer (perhaps multiple manufacturers in parallel)	Product Type Certificate (TC)
3	Introduction of the specific RPAS product into service in a specific region	Air Operator	(Updated) Air Operator Certificate (AOC)

Table 3: Illustration of development with stages spanning the system lifecycle

This approach is already common practice within the industry.

Note that there may be different *change leaders* at different stages of the lifecycle. (At each stage the *change leader* will be supported by other organisations.) Each *change leader* will have slightly different goals, albeit within the overall context of introducing the system to service.

It is possible to apply the ASCOS Method to each of the separate stages in this case, but the details of the *safety argument* will be significantly different, reflecting the stage in the lifecycle. The *approval* required in each case will also be different: in fact, for stage 1 there may be no provision for formal *approval* in the regulations – however, some form of *approval* of the requirements generated would be expected by the industry before they develop products to meet the requirements.

4.5.2 Stages reflecting different operational states of the final system

Deployment of a new solution may go through a number of interim operational states. For example, when introducing a new arrivals concept at an airport, which involves reducing separation between aircraft, the airspace changes (sectors and routes) may be introduced first and proved in operation, before actually reducing separation. Alternatively, a new air traffic management solution may be deployed progressively as the existing equipment reaches the end of its design life, or it may be deployed to multiple areas, constrained by the budget of the ANSP.

Each operational state will need to be addressed separately in the *safety argument* to demonstrate that the operation in that state will be acceptably safe. As well as justifying the safety of the state itself, the transitions between states must be assessed to ensure that each transition can be completed safely. This transition assessment is needed even when there is a single transition (from initial state to final state), but becomes more involved and critical where multiple changes are made in a relatively short period of time.

In this case, it is likely that the same organisation will be responsible for each part of the *safety argument*.

5 Safety Argument for Aviation Changes

5.1 Introduction to *Safety Arguments*

The term *safety argument* is used to refer to a *logical argument* which makes a claim that a system achieves an *acceptable level of safety*.

A *logical argument* is a connected series of statements, with supporting evidence, used to persuade the reader of the correctness of an overall *claim* or conclusion. It is not an argument in the sense of a disagreement. A *logical argument* attempts to supply strong evidence, rather than absolute proof, of the truth of the *claim* being made. In the case of a *safety argument*, the *approver* will usually require very high confidence to be established in the *claim* being made, and hence the *safety argument* must be rigorously constructed and reviewed. *Logical arguments* are susceptible to a number of pitfalls which can undermine the argument: more details of possible pitfalls are discussed in section 5.5.

Safety arguments have been accepted across a range of industries for over 15 years as a means of enabling clear, concise and traceable arguments for safety assurance to be presented to regulators. A brief summary of some uses of *safety argument* in this way is presented in Appendix D.

5.1.1 What is a *Safety Argument*?

A *safety 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*). Evidence can be categorised as
    - direct evidence - this is evidence, relating directly to observable properties of an output or product, that a particular objective has been achieved
    - backing evidence – this is evidence that there is sufficient confidence that the direct evidence can be relied upon; backing evidence relates to properties of the processes by which direct evidence was obtained: e.g. tools and techniques, human resources applied were qualified/competent and properly deployed
- Caveats (*limitations*, *constraints*) which constrain or limit the interpretation and further application of the *claims* made
- Dependencies on other components outside the bounds of the *change* under consideration

An *argument* is presented as a hierarchy below a top level *claim*, usually of the form “System X is acceptably 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.

The supporting information (*strategies*, *context*, *justifications* and *assumptions*) is critical as it explains the *safety argument* and makes clear any parameters within which the *safety argument* was constructed. This is especially important when reviewing the *safety argument*, or attempting to re-apply an existing *safety argument* to another *change*.

Safety 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)

5.1.2 Why do we need a *safety argument*?

Construction of a *safety argument* requires rigorous and detailed examination of the *claims* being made and the evidence available to support them. The exercise of constructing the *safety argument* therefore requires the organisations seeking to make the *change* (the *change leader* and the *applicant*) to think carefully through the claim being made. This attention helps to improve the validity of the *safety argument* and provide confidence in its conclusions, before it is submitted for *approval*.

A further purpose of the *safety argument* is to demonstrate that a proposed *change* will be acceptable and to communicate the reasons for that belief to interested stakeholders, in particular the *approver*.

The *safety argument* demonstrates that the proposed *change* will achieve an *acceptable level of safety*: the *safety argument* covers all modes of operation, including fall back; it also includes the, transitional stage(s) required to implement the *change*.

All *approvals* are based on a *safety argument* of some form. This may be an implicit argument effectively defined by procedures to be followed to gain *approval*, or it may be an explicit argument presented in *approval* submissions, e.g. by constructing a safety case. In some *domains* the *safety 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.

A well-structured explicit *safety argument* provides the mechanism to argue that a *change* can and will be implemented, in a compelling and comprehensible way. A complete and correct *safety argument* can ease the *approval path* as it should provide clear pointers to evidence in support of a top level *claim*. A complete *safety argument* is one that covers all relevant aspects of the *TAS* to a sufficient and necessary level of detail. A correct *safety argument* is one that:

- accurately reflects the design
- is consistent
- is both logical and understandable
- is supported by empirical evidence, proof or reason

Even within these constraints, different types of *safety argument* may be constructed. It would be possible to create a separate *safety argument* for each requirement within a specification. However, the *safety argument* envisaged by the ASCOS Method is more of the following form.

- The relevant standards are sufficient to assure that the *acceptable level of safety* is achieved.
- The *change* has complied with these standards.
- The context of implementation of the *change* matches the context envisaged by the standards applied.
- Therefore, the *change* is adequately safe.
- The *safety argument* is accepted by the relevant *approver*.
- Thus, by implication, the *change* achieves the *acceptable level of safety*.

A *safety argument* of this latter form is much smaller (and therefore easier to construct and understand) than one individually justifying compliance with each requirement.

5.1.3 Explicitly Stated Arguments and Assumptions

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 *safety argument* is fallacious.

An example is the changing role of an aircraft's Flight Management System (FMS) with the introduction of advanced functions such as advanced RNP: the *safety 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.

Another example comes from the adoption of composite materials in airframes. Certification Specifications (CSs) specify the requirements which must be met in order to obtain a Type Certificate for an aircraft: they therefore form part of a compliance based *approval* approach. The parts of the CSs which relate to physical structure are generally not specific to a particular type of material. However, as metallic structures have been the norm in airframes for many years, and because the CSs have been developed over this same timeframe, the CSs often (implicitly) assume that the airframe will be constructed largely or entirely in metal. Some of these embedded assumptions are easily apparent, e.g. references to corrosion. However, others are less obvious. For example, the mechanisms for growth of cracks in metal structures mean that some cracks can be tolerated, as long as they are detected and monitored. However, damage growth in composites can be rapid, unpredictable and not readily detectable, meaning that a very different approach is needed for composites compared with metal structures.

This particular issue has been recognised by EASA and addressed in guidance information [10] and the relevant CSs (e.g. CS-25 [8]) have been subsequently updated to take the use of composites into account. This is an

example of where review of context and assumptions has led to revision of the specification on which *approval* (in this case *certification*) is based.

This example illustrates the importance of *context* and *assumptions* in developing an *approval* path and subsequently the *safety argument*. Where the *approval* path is based on any existing *safety arguments* or evidence, the relevance of these *safety arguments* to the *change* under consideration needs to be carefully evaluated. In this example, attempt to make a *safety argument* for a composite airframe based on an older version of the relevant CS, should have identified that the CS (implicitly) assumed metallic structures. This should then have led to a review of the CS to identify how the move to composites affected the requirements. The organisation driving the *change* would then need to decide whether to develop a specific *safety argument* to address the elements of the development not (fully) covered by the CS or to support / request redevelopment of the CS to support the use of composite materials.

Further details are given in a paper [11] in the Journal of Aviation Management 2014.

5.1.4 How to present a *Safety Argument*

A *safety argument* may be presented in a variety of implicit or explicit forms often purely textual or compliance based.

Explicit use of a graphical notation with a formal syntax helps both the author and the reviewer by encouraging thorough consideration of the logical structure and justification of the *safety argument*. It thus 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 *safety 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 [12] and supported by multiple research papers and presentations. It is also described in the EUROCONTROL Safety Case Development Manual (SCDM) [13] and in a UK CAA guidance document on production of safety cases [14], although these documents do not specifically recommend any particular graphical approach.

GSN is chosen over other graphical notations for presentation in the ASCOS Method 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

- it is covered in industry publications as described above

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

5.2 A *Safety Argument* for Aviation

5.2.1 Why This *Safety Argument*?

The ASCOS Method has adopted a generic *safety argument* which is presented in section 5.2.2. There is no single "correct" *safety argument*. Other arguments are possible, but the *safety argument* presented here has been used successfully in aviation applications (see Appendix D) and has been refined through that use.

The *safety argument* presented here is aimed towards extensive, multi-*domain changes* to the *TAS*, where the degree of innovation requires significant adaptation to existing approval paths.

It should be noted that, in the ASCOS Method, the purpose of the *safety argument* is to support the chosen *approval path*. The concept of the *approval path* is explained in section 3. A *change* which was simpler, or which was more capable of being approved using existing approaches, would be able to use a much simpler *safety argument* than that presented in section 5.2.2. For example, where the *change* largely adopts an existing *approval* path, with a minor adaptation, it would only be necessary to make a *safety argument* for that adaptation to the approval path (as illustrated in Figure 5).

Whatever *safety argument* structure is used, it is critical to ensure that the *safety argument* addresses the whole system lifecycle: it is not sufficient just to demonstrate that a particular design has been implemented, it is also necessary to demonstrate that the design sufficiently addresses the intent of the *change*, and that the *change* is monitored in operational service to confirm that the *change* to the *TAS* does indeed achieve the *acceptable level of safety*.

5.2.2 A Generic *Safety Argument*

Figure 14 presents a generic *safety argument* for use within the ASCOS Method. This *safety argument* is based on a generic argument which is already widely used within ATM. Originally developed within the EUROCONTROL Safety Case Development Manual (SCDM) [13] and subsequently extended and enhanced by the SESAR research programme (see [15]) this generic *safety argument* addresses all stages of the development lifecycle, from concept through to maintenance in continued operation.

This *safety argument* is intended as a template to be adapted according to the needs of the change, as discussed further in section 6; the argument here is biased towards the needs of a substantial multi-domain change where significant changes to existing approval paths need to be justified.

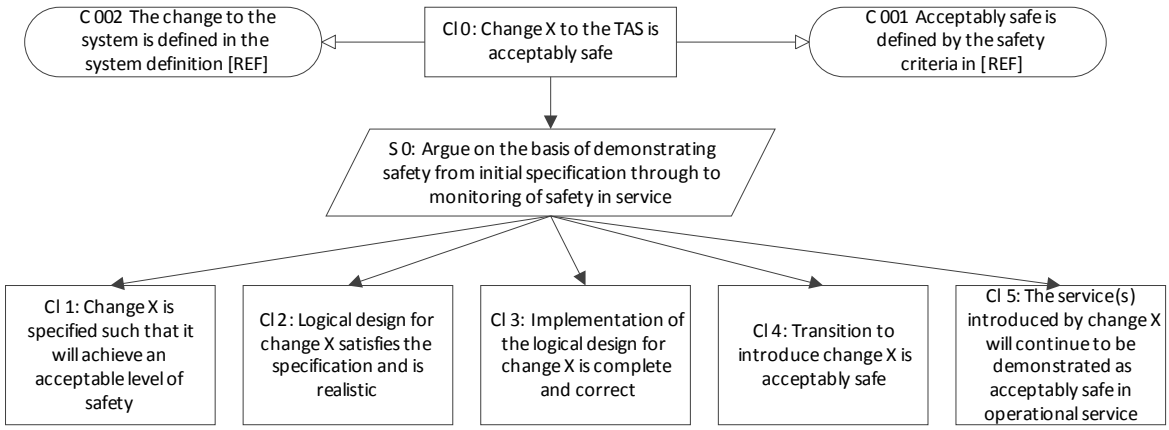

Figure 14: Generic Safety Argument

The *argument* starts with the top level *claim* (Claim 0: "Change X to the TAS 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 (C 002)
- definition of the term "acceptably safe", through definition of safety acceptance criteria (C 001)
- *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

Note: only the first two items are shown in this example, but all of these types of *context* would usually appear at some level in the *safety argument*.

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

5.2.3 Developing the *Safety Argument*

The *safety argument* needs to be developed so that the top level *claim* (that the *change* achieves the *acceptable level of safety*) can be supported by appeal to evidence in some form (see below). This section gives guidance on developing the *safety argument* by decomposing the top level *claim*. Partitioning the safety argument across the *domains* of the *TAS* is dealt with separately in section 5.3. Further, more general, guidance can be found in the papers referenced in Appendix D. Note: detailed guidance on decomposition of *safety arguments* is difficult as, by its nature, it is a creative exercise.

Each *claim* of the *safety argument* is decomposed into sub-*claims*, each addressing a part of the parent *claim*, such that when the sub-*claims* are taken together they completely address the parent *claim*. Decomposition of

claims should be supported by *strategies* which explain the approach taken in the decomposition. Use of *strategies* helps to explain the *safety argument* and assists reasoning as to the completeness and correctness of the *argument*. *Strategies* should be supported by additional information as necessary to ensure that the *safety argument* is clearly stated; this will include any *assumptions* made by the *strategy* and any *justifications* required to demonstrate that the *strategy* is valid.

The *safety argument* should make the link between the top level *claim* and the evidence produced during development of the *change*. Where the evidence produced by existing approaches (e.g. standards, AMCs) is sufficient to support the claim, the *safety argument* should only be developed down to the execution of that approach, along with justification that the context (see section 5.1.1) assumed by the approach matches the context required by the *safety argument*.

Where existing approaches do not provide the evidence required to support the claim, either the existing approaches must be adapted or augmented, or it may be necessary to develop a new approach to generate the evidence needed. (Where possible, these new approaches should be developed in such a way that they can be reused in future applications.) Where new or adapted approaches are developed, it may be necessary to develop the *safety argument* in more detail in order to justify these approaches and to ensure that the generated evidence will be sufficient. Especially where new approaches are developed, the *safety argument* should be specific about the evidence required and why it is required; this supports the exercise of reviewing the *safety argument* to determine whether its *claims* are in turn satisfied by that evidence.

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.

At all levels of the safety argument, it is important to consider both direct evidence and backing evidence as explained in section 5.1.1.

5.2.4 The Next Level of the Argument

The top level claim (Claim 0) is broken down into the *claims* shown in Figure 14 which address the different stages of the development lifecycle[14]. The development of these claims is further addressed in section 6.5.2.

- **Claim 1: Change X is specified such that it will achieve an acceptable level of safety**: This *claim* focuses on <u>what</u> is being changed (e.g. introduction of a new concept or service) without considering the details of how the *change* is implemented. In this claim, the *change* is considered in terms of high level functionality and performance, operational behaviour, modes of operation and scenario analysis. Even at this level, the *change* should be partitioned into the different *domains* within the *TAS*

[14] Alignment to system development lifecycles is discussed in section 4.4.

to facilitate initial development of the *safety 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.

- **Claim 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 Claim 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.
- **Claim 3: Implementation of the logical design for change X is complete and correct**: This *claim* demonstrates that the 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 actual equipment and operations are made.
- **Claim 4: The transition to introduce change X is acceptably safe**: This *claim* is concerned with assuring that the components of the change (equipment, people and procedures) can be safely brought into operational service, considering both the readiness of the components and the safety of the transition itself: this includes assuring that the *change* can be brought into service without adversely affecting the safety of existing on-going operations during the period of transition from current operations to the new situation. Where a *change* is introduced in multiple stages, each individual stage needs to be fully considered within this *claim*.
- **Claim 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 Claims 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 *external environment* is critical: this is necessary so that

[15] In this context, logical design is a high-level architectural representation, independent from the implementation. As such it considers the functions provided by system elements (i.e. human roles and tasks and machine-based functions), but not 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.
[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.

the correct items in the *TAS* and the *external environment* can be monitored and so that corrective action can be taken where necessary.

5.3 Partitioning the *Safety Argument*

Safety 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 *safety 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, complete and correct overall *safety argument*. This approach is analogous to similar principles in software and system design.

5.3.1 *Safety Argument Modules*

A *module* encapsulates a particular part of the *safety argument*; for example *modules* could be used to divide the *safety argument* into the individual *domains* of the *TAS*, or to contain the parts of the *safety argument* pertaining to individual organisations. (More guidance on the choice of *modules* is given in section 5.3.3.)

A *module* defines a number of *claims* made by the *module* and provides the *safety argument* structure (potentially including all the types of element of a *safety argument*) to support those *claims*. The *safety argument* made by the *module* will have associated *context* and *caveats* and will have dependencies on *claims* made outside the *module*.

The interface of a *module* should be “public” in the sense that a number of attributes are defined to allow the *module* to be composed with other *modules* to form a complete *safety argument*. The detail of the *safety argument* in the *module* can be “hidden” within the *module*, although it obviously must be available for review to confirm that the *claims* made by the *module* are demonstrable. The “public view” of a *module* is illustrated in Figure 15.

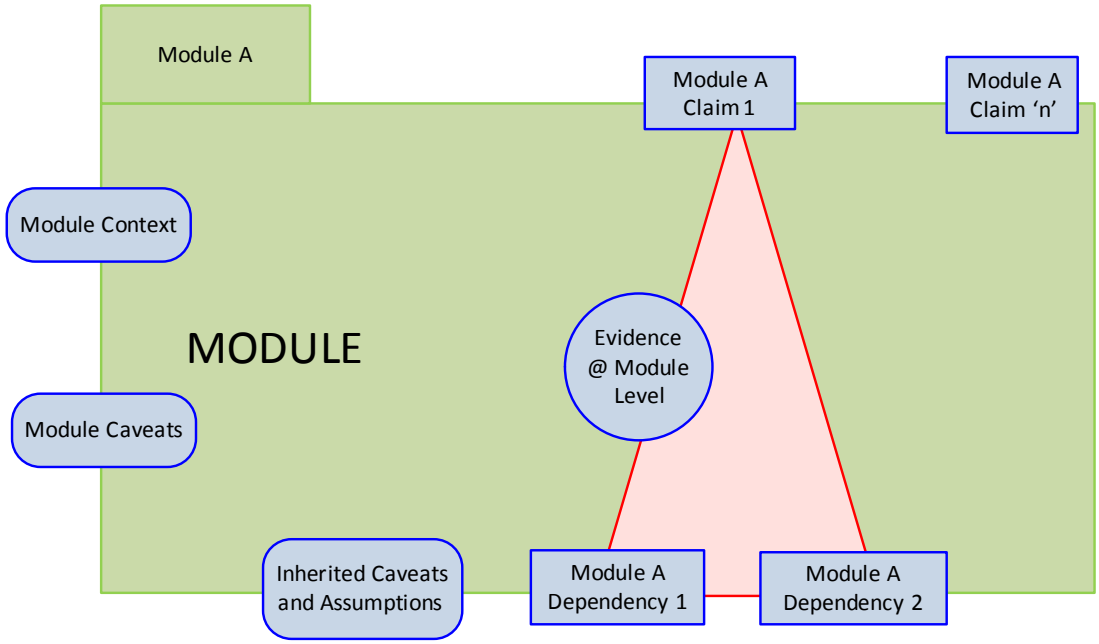

Figure 15: Public View of a Safety Argument Module

The attributes defined at the boundary of the *module* shown in Figure 15 are:

1. *Claims* made by the *module* – i.e. those which the *module* provides the *safety argument* to support
2. *Module context*, defining the environment within which the arguments in the *module* are developed – see section 5.3.1.1
3. *Module caveats,* defining the parameters which must be respected in order for the *safety argument* in the module to be valid – see section 5.3.1.1
4. *Dependencies - claims* identified within the *module*, but for which another module provides the argument to support
5. *Inherited caveats and assumptions*, imported from other *modules*

In addition, the following may also be defined at the boundary of a *module*, although not shown here:

6. *Evidence* presented by the *module*
7. References to *evidence* presented in other *modules* which is required to support the current *module*.
8. References to *context* defined in other *modules* which forms part of the *context* for the current *module*.

Clear definition of the *module* interface 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.)

Modularisation is important to the ASCOS Method because it allows:

- Clear definition of the interaction between different elements of the *change* and of the *TAS*.

- Compartmentalisation of different parts of the *safety argument*, allowing updates to parts of the *safety argument* without affecting the rest, as long as the *safety argument* is unchanged at the interface of the *module*.
- Reuse of parts of the *safety argument* without requiring extensive redevelopment, again as long as the *assurance contracts* with other parts of the *safety argument* remain unchanged.
- Individual parts of the *safety argument* to adopt the practices habitually used in the *domain* while also ensuring that the *safety argument* can be integrated with the rest of the *TAS*.
- Alignment of *modules* to *domains* of the *TAS* to simplify *approval*.

5.3.1.1 Module *Context* and *Caveats*

The *context* and *caveats* published at the boundary of the *module* are key elements of the modularisation.

The *context* defines the environment (in the widest sense) in which the *safety argument* is made.

The *caveats* define items which must be considered when applying the *safety argument* because they either restrict the way in which components of the *change* can be put together, or they restrict the operational service use to which the *change* can be put.

It is therefore critical to ensure that the *context* and *caveats* are correctly published at the module boundary, because the *safety argument* made by the module is only valid if:

- the *context* reflects the environment within which the change is used
- the *caveats* are respected

An example is given in the next section.

This highlights the need for someone (the *argument architect*) to consider the overall *safety argument* and ensure that the separate *modules* are correctly integrated – see section 5.4.

5.3.2 Assurance Contracts

The primary links between modules are dependency-claim relationships, the dependency of one *module* (for example the assurance of software elements of the change) is linked to the claim of another *module* (i.e. that a particular software package is assured to a defined assurance levels)[18]. However, each *claim* also has associated *context* and *caveats*. Part of the *module* linking process is to ensure that the *claims* are valid in the *context* relevant to both *modules* and that any *caveats* are correctly transferred. In the above example, it may be necessary to ensure that the software has been assured to operate in the environment (e.g. on the processing platform) used on the aircraft.

[18] Note there may be many links between the same modules and these can be rolled into a single link to avoid over-complication.

Assurance contracts[19] provide a way of documenting and managing these links between *modules*. The *assurance contract* provides a means of justifying that the dependencies of one *module* are satisfied by the *claims* in another and ensuring that the related *context* and *caveats* are correctly communicated between *modules*. The example described above is illustrated in Figure 16.

Where *modules* are used to capture existing *safety arguments*, such *assurance contracts* may already partially exist as agreements between domains in the form of interface standards. Such standards should be used where possible, to prevent redeveloping interfaces which already exist. However, these standards need to be evaluated to confirm whether they cover all the aspects required by the *assurance contract* and are valid for the required context, or whether additional agreements need to be developed.

[19] 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.

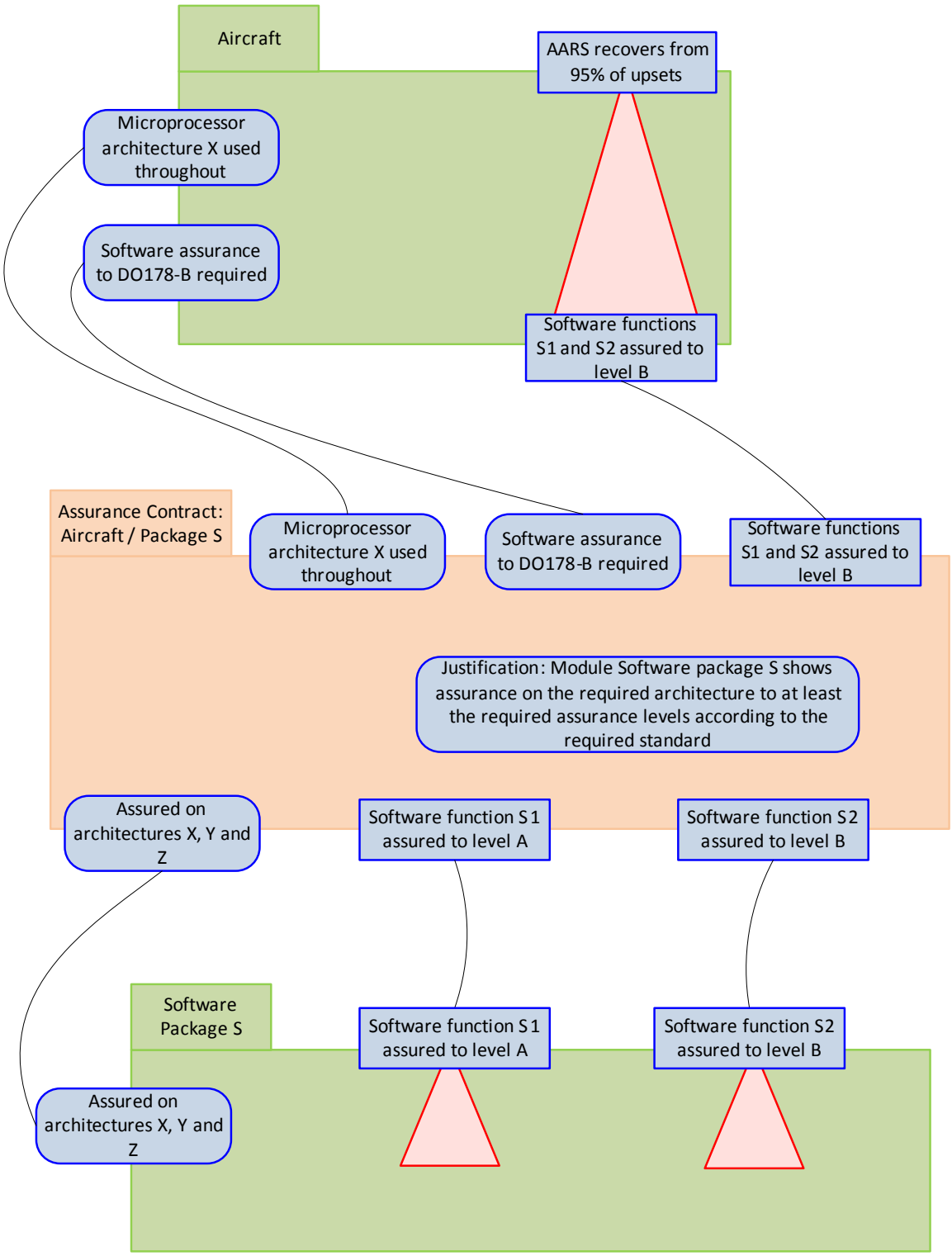

Figure 16: Illustration of linking modules using assurance contracts

5.3.3 Choosing *Safety Argument Modules*

Systems theory dictates that successful modularisation depends on modules being loosely coupled and highly cohesive. One specific reason for use of modularisation within the ASCOS Method is to partition the *safety argument* into *modules* where approval will be granted by different *approvers*. There are also other reasons for the choice of *modules*: they may align to:

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

In addition, modules may be used:

- 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 *approval* 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 *change*
- 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*
- as a container for the argument relating to backing evidence (see section 5.1.1), where this evidence may by used in multiple locations throughout the *safety 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 *safety argument* where necessary
- as a container for volatile parts of the *safety argument* in an attempt to minimise the effect of this volatility on other parts of the *safety argument* – obviously the key to this success is the ability to define a stable interface for the module (Management of variations is discussed further in section 6.8.)

5.3.4 Limiting the Effects of Variation

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 modified - this could be for one of a number of reasons, including:

- variation of the *change*
- part of the *safety argument* is found to be incorrect
- inability to provide the *evidence* envisaged when the *safety 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 modification to a single *module*, or a small number of *modules*. Although it will still be necessary to repeat the verification step, this should also be simpler, as only the items which have been modified (and their effect on other items) need to be revisited.

In addition to careful choice of *module* boundaries, 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 *safety 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 *safety 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 *safety argument*, with the potential side effect of making that *claim* more easily reusable in other parts of the *safety argument*.

5.3.5 Example *Module* Structures

As the example 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.

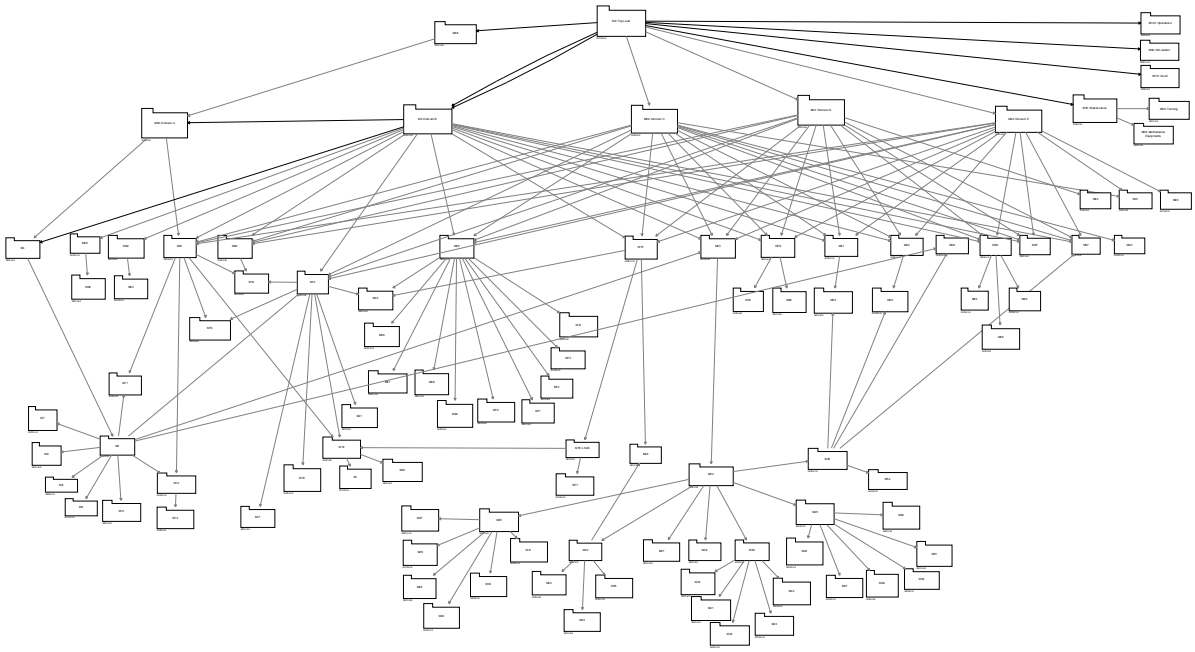

Figure 17: Example modular safety architecture

Note: the diagram is only intended to be illustrative, to highlight the complexity of the assurance interactions that can be present in a typical safety critical system. In this illustration it is not intended that the detail in the boxes should be readable.

Figure 18 presents a possible *safety argument* architecture for the safe operation of Electronic Flight Bag (EFB) technology. 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.

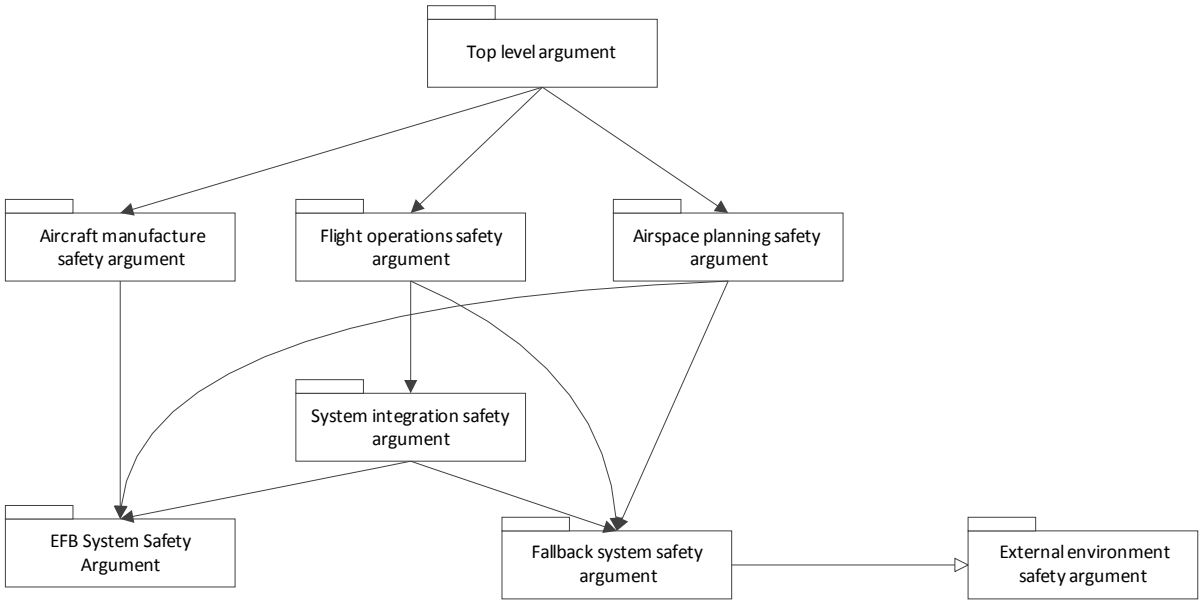

Figure 18: Modular Safety Argument Architecture for Operation of Electronic Flight Bag (EFB)

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 airspace planning, a *module* is still required for the airspace planning *safety 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 *safety argument* about this environment.

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

5.4 The Need for an *Argument Architect*

It has been shown above that modularisation allows subdivision of the *safety argument* into *modules* connected by *assurance contracts* so that these *modules* can be developed separately from one another in confidence that the final result will be a consistent, complete and correct overall *safety argument*. However, this also introduces a significant risk of divergence between the *modules* in ways which were not envisaged when the *modules* were created. It is therefore necessary to ensure that the *safety argument* is properly maintained and integrated throughout the development.

When engineering complex systems, the role of system architect is responsible for the design of the overall system; this includes ensuring the integration of the resultant modules. The ASCOS Method gives the *argument architect* the similar role of designing and maintaining the *safety argument*, which includes ensuring that the *safety argument modules* are correctly bounded and interfaced to other *modules* throughout the development.

When considering the number of organisations involved in the *TAS* and their disparate roles, it is often not easy to identify who should be the *argument architect*. This in part explains why a key concern within the industry is the inadequacy of the management of interfaces between *domains*; sometimes integration is supervised by the *approver* or even ignored altogether.

Whilst the *approver* is in a position to oversee the *argument architect* role it would be inappropriate to task *authorities* with engineering the integration. Due to the way in which such *safety arguments* span multiple domains, it may not be possible for whole *safety argument* to be approved by one *approver*. As a result, it is necessary during the initial planning of the *approval* approach to clearly define the parts of the *safety argument* which require endorsement by each *approver*.

(Section 7.1.4 discusses the *argument architect* role in more detail, and the relationship with the *TAS Engineering and Safety Group (TESG)*.)

5.4.1 Responsibility for Maintaining the *Safety Argument*

The *module* owner remains responsible for the content of their individual *module*(s) of the *safety argument*, including ensuring that the *module* is correctly represented at its interface. (This includes ensuring that the *module* does indeed demonstrate the *claims* which it makes as well as ensuring that all the relevant *context*, *caveats* and dependencies are correctly stated.)

The *argument architect* is responsible for ensuring that the *modules* are interfaced correctly via *assurance contracts*. This will require co-ordination with the *module* owners and *approvers*: it may be necessary to implement this co-ordination via the *TESG* (see section 7.1.5).

5.5 Problems and Pitfalls

The *safety argument* approach is not without its critics. A recent example is the Haddon-Cave investigation into the loss of a Nimrod aircraft over Afghanistan [17] which levelled a number of criticisms at the use of safety cases. This is not a criticism of the use of *safety 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[20] should be:

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

A rigorous approach to *safety arguments* helps to achieve these objectives. Some ways in which this is achieved is through ensuring that the *safety argument* remains focused on the goal and by ensuring that the *safety argument* is structured using precise definitions so that it is easy to follow and to reason about. *Safety arguments* must not be made more complex than necessary[21], and should adopt existing approaches (such as demonstration of compliance with existing standards) at the highest level possible.

Care is certainly needed when constructing a *safety argument*. Mistakes can be made or poor reasoning can be used in the construction of *safety 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.

[20] Haddon-Cave also recommended that the documents should be renamed “Risk Cases”.
[21] However, complexity of the *safety argument* is often driven the by complexity of the system (in the widest sense of the term) about which the *safety argument* is being made.

The OPENCOSS deliverable D4.1 [18] (section 3.3.3) contains a summary taxonomy (first published in [19], 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 [20] and [21] for examples of some of the issues with circular arguments and appeals to expert judgement. Categories of fallacies are:

- 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 which establishes (or counters) the validity of the safety argument.
- 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 the 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 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.

6 Applying the ASCOS Method

6.1 How to use this section

In section 3, the ASCOS Method was described at a high level as a sequence of steps, as illustrated in Figure 19. The following sections (section 6.2 to section 6.7) provide more detailed guidance on how to apply each of these steps to a specific *change*. It should be noted that this description is not intended to define a formal process; instead it should be treated as a framework to be used to develop an *approval path* (with supporting *safety argument* and *approval plan*) for a specific change.

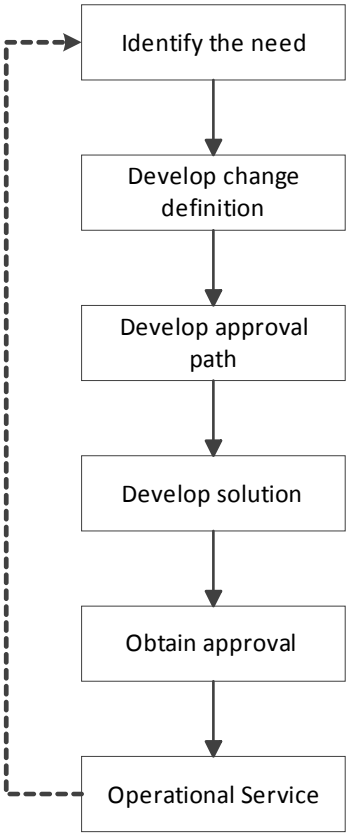

Figure 19: Overall View of ASCOS Method (copy of Figure 2)

The development of a *safety argument* is used where necessary to demonstrate that the chosen *approval path* achieves the *acceptable level of safety*. The *safety argument* is divided into *modules* aligned to the *domains* of the *TAS*, with *assurance contracts* defined between the *modules* to ensure that the dependencies between parts of the *change* are correctly captured and managed.

The ASCOS Method is shown as a linear sequence of steps. However, as noted in section 3.1, a degree of iteration is usually required. The implementation of this iteration is discussed further in section 6.8.

The ASCOS Method is designed so that it can be used alongside a range of system development lifecycles. As an example a mapping to the E-OCVM [6] lifecycle is given in Table 2 (see section 4.4). The organisation applying the ASCOS Method should map its steps to their own lifecycle; this will then provide a guide as to when the steps need to be undertaken. Note that the steps of the ASCOS Method span multiple stages in the development lifecycle.

The concepts of the ASCOS Method are intended to be widely applicable. In particular it is not limited to *certification* where an item (or organisation) is confirmed to comply with an agreed standard; instead it can also be applied to *approval* where there is not an agreed standard against which to measure the subject of the approval. However, it is also intended that the ASCOS Method can be applied in the case of *certification* if it is useful to do so – for example where the proposed change is mostly, but not entirely, covered by existing standards.

The ASCOS Method can also apply to *changes* which do not map directly onto the stages shown in Figure 19. For example, development of a new regulation or SARPS is only the first stage in developing a novel *change* and putting it into service: the regulation itself does not enter operational service. However it is still possible to use the ASCOS Method to develop such a regulation, but the steps of the method will need a degree of reinterpretation. Note that the feedback cycle still applies, as application of the regulation may uncover flaws which need to be addressed in amendments.

Another *change* for which the lifecycle may appear significantly different is the licensing of a new organisation. The ASCOS Method is still applicable: in this case the *change* definition should be understood as the definition of the remit or scope of operation of the organisation; development of the solution should be understood as designing the organisational structure and the procedures to be followed.

The following sections present a logical progression through the application of the ASCOS Method. Iteration is addressed separately in section 6.8.

6.2 Identify the Need

The *approval* process really starts with defining the change. However, before the *change* to the *TAS* can be defined, it is necessary to identify that the *change* is needed, and it is useful to understand this step in order to appreciate the context for a *change*. However the *approval* process truly commences with the definition of the *change* as described in section 6.3.

Needs for *change* arise because of:

- a business need
- continuous safety monitoring
- external changes

Note: changes in one part of the *TAS* can lead to changes in another part of the *TAS*: these should be treated as a single *change* to ensure that the impact of a given *change* is fully understood and addressed.

The organisation identifying the need will be different in each case. Furthermore, the bullets above represent broad groups of changes: different *changes* within each of the groups will have different *change leaders*. Each of these broad needs for *change* is discussed further in the following sections.

Once the need has been identified, *changes* can be developed to address the need. A process of exploring the possible *changes* needs to be undertaken to determine what (if any) *changes* are feasible. Any organisation will have its own established process for making the required business case for a *change*; construction of business cases is outside the scope of the ASCOS Method. However, an understanding of how the ASCOS Method will be applied is necessary as an input to establishing the business case.

6.2.1 Business Needs

The business need for a *change* may include:

- additional capacity within the TAS (due to demand exceeding capacity)
- greater efficiency (because of increased costs or reduced revenue)
- replacement of an obsolescent part

These needs are likely to be identified by operators, when they find that the existing system is

- operating at maximum capacity (e.g. unable to accommodate the number of passengers who are requesting service) or
- operating inefficiently

Organisations within the TAS may identify *changes* to their own processes or they may identify possible *changes* to the wider system, which they may pass on to their suppliers. The type of *change* proposed could be:

- development and introduction of a new (perhaps larger) aircraft
- a new operating concept (e.g. RPAS operating in unsegregated airspace)
- new operating arrangements (e.g. RVSM or self-assured separation)
- a new operator providing low cost flights on existing routes
- a replacement part based on new technology because the previous part is no longer available

6.2.2 Continuous Safety Monitoring

Continuous safety monitoring (CSM) is an essential part of any safety management system (SMS) to monitor the level of safety achieved by the system and to identify any trends which indicate degradation of safety. EASA undertakes some monitoring at the overall European level, which results in actions which are published in its European Aviation Safety Plan (EASp) [22]. This is complemented by requirements within the relevant regulations for provision of services (e.g. ATS.OR.200 (a) (3) (i) within the draft regulation for (air traffic) service providers [23]).

ASCOS has developed a methodology and supporting tools for multi-stakeholder CSM, using a baseline risk picture for the *TAS*. (This is documented in the final report for Work Package 2 [7].) This methodology identifies safety performance indicators (SPIs) which can be monitored to reveal trends and therefore indicate areas where change is needed in order to maintain or improve safety.

When a *change* is introduced, the *safety argument* for the change will be based on predictions of safety performance. Part of the *safety argument* includes putting in place monitoring to show that these predictions are achieved in practice. The general SPIs should be augmented by SPIs associated with individual changes to the TAS, where additional parameters need to be monitored. These additional SPIs are developed as part of constructing the *safety argument* for an individual *change*.

Where CSM reveals trends of decreasing or unacceptable safety, this indicates a need for change to the *TAS*[22]. This could be implemented through development of a safety directive by the *authority*. However, it will often be the case that the development and implementation of the *change* will require involvement from organisations across the *TAS*.

6.2.3 External Changes

A *change* in the environment outside the *TAS* may lead to changes being needed within the *TAS*. Examples could include:

- changes to external regulations
- changes to prevailing weather patterns or extremes of weather
- construction of new tall buildings
- changes to patterns of military activity

In each case the external change would initially be identified by the organisation affected by the change, which would be responsible for evaluating the need for change to the *TAS*.

[22] In the case where the CSM is monitoring the safety of a previously implemented *change*, this could be considered as a modification of that *change*, but in practice it is better to consider it as a new *change*.

6.3 Define the *Change*

Once the need for *change* has been identified, the *change* itself needs to be defined. This definition is an iterative process, especially during the early parts of the lifecycle. Multiple *changes* may be proposed and evaluated: some will be discarded as not meeting (all) the need, or as being infeasible. The *change* definition will become more detailed as the lifecycle progresses.

The ASCOS Method is not intended to replace existing methodologies for defining the *change*. Instead, this section identifies what aspects of the *change* need to be defined so that the ASCOS Method can be applied. If an existing methodology does not deliver all the aspects covered here, it can still be used – but it needs to be extended to ensure that all these aspects are covered.

The following aspects of the *change* need to be identified, in order to apply the ASCOS Method:

- functional definition
- influences within the existing system shaping the *change*, including future anticipated *changes*
- assumptions
- impact of the *change*
- stages of the *change*
- transition into service
- organisations involved
- the *acceptable level of safety* which the *change* needs to achieve

Each of these aspects is covered further in the sections below.

Once the aspects above have been defined, the *approval path* for the *change* can be developed: this is covered in section 6.4.

If later alterations are made to any aspect of the definition of the *change*[23], the overall *change* definition should be reviewed and updated accordingly. The impact of this alteration on the *approval* path (see section 6.4) and the later steps of the ASCOS Method must also be assessed and updates made accordingly.

6.3.1 Functional Definition

The *change* must be defined in terms of function and performance. In other words, it is necessary to answer the question: what will change about how the *TAS* operates or behaves? This must go beyond "what to we want to achieve?" and define the *change* to the *TAS* at a conceptual level, so that the affected *domains* and applicable regulations can be identified.

[23] This is almost inevitable with complex changes.

For example, if we want to achieve a reduction in loss of control incidents, possible solutions include:

- development of an Automated Aircraft Recovery System (AARS)
- improved pilot training in upset recovery

It is clear that development of an AARS will have significantly more technological involvement, including greater impact on other domains such as ATM, than choosing improved pilot training.

However, it is not necessary at this stage to define the detailed implementation of the change. A common mistake at this stage is to descend deep into the detailed design of the *change* (e.g. the architecture of the equipment involved) without fully defining the functional changes to the *TAS*.

A key goal of creating the function definition is to allow all the *hazards* at this conceptual level to be identified.

It is also important to develop a description of how the *change* would be used within the *TAS*: this is known as an operational concept and should cover both normal (intended) operation and operation in abnormal or degraded conditions. (The development of an operational concept is covered in various standards and methodologies within the industry, such as E-OCVM [6] and DO-264 [24] Annex C.)

For example, introduction of an automated aircraft recovery system (AARS) could be defined as a function which allows the pilot to request automated recovery of the aircraft to stable, level flight. The description would then be developed to work out how this pilot would interface with this function, and how the function would affect other *domains* within the *TAS* (e.g. the air traffic control domain). It would also describe the scenarios in which the function would be used. However, at this stage, it is not necessary to describe the technology which would be used, nor where it would be placed in the cockpit – these details can be introduced at a later stage of development.

6.3.1.1 Generic vs Specific *changes*

Where the *change* is an operational *change* to the *TAS*, it is important to define the actual *changes* which are proposed. For example, where an operator is proposing introduction of a RPAS into unsegregated airspace, it is important to specify features such as:

- the types of flight proposed - e.g. will it fly from one aerodrome to another (to deliver goods) or will it hover / circle over a particular area (to perform some form of observation)?
- the sectors which the RPAS will fly through
- the functions which the ANSPs will provide in relation to control or surveillance of the RPAS

It is also useful to identify the specific organisations involved (e.g. ANSPs, maintenance organisations) as these will need to be consulted regarding the impact of the change on their operations.

These details are important because they significantly affect the *hazards* which may be introduced or affected by the operation, and the available mitigations for them.

However, many *changes* start off as development of a new product or concept, which later becomes a specific *change* to the *TAS*. With such *changes*, it is not initially possible to describe a specific operational *change* to the *TAS*. In this case it is crucial to define the *assumptions* made about how the product or concept will be operated. In order to make the product or concept as broadly applicable as possible, the manufacturer will want to make these assumptions cover a wide range of operations. Assumptions are covered further in section 6.3.3.

6.3.2 Factors Shaping the *Change*

The existing *TAS* has a large effect on the *change* being made.

- The *change* must operate within the structures of the existing *TAS* (except where those structures themselves are being changed).
- The *change* must consider the relevant existing regulations. In some cases the ASCOS Method will be applied to make a *safety argument* to replace demonstration of compliance with existing regulations. However, there may be other regulations with which the *change* will still need to comply.
- The scope of the *change* will be limited by what it is feasible to *change* within the existing TAS.
- The scope should be restricted to only that which it is necessary to *change*: existing operations should not be changed unnecessarily – this will introduce additional cost and disruption.

All these factors may affect the *approval* path and the *safety argument* for the change.

It is also important to take into account other foreseen changes. The Future Aviation Safety Team (FAST) Areas of Change (AoC) list [25] presents a list of expected changes to the worldwide aviation system along with related *hazards*. This list should be reviewed during *change* definition to identify areas which have an impact on (or are impacted by) the proposed *change*. The *change* should take the impact of these future changes into account, avoiding the need for a further adaptation of the *change* when items in the AoC list are introduced into the *TAS*. For example, introduction of an RPAS will be significantly affected by AoC_11, which notes the increasing diversity of aircraft fleets, in terms of size, capability and equipage.

6.3.3 Assumptions

During the development of a *change*, it is often necessary to make *assumptions* where the facts are not known. This is especially true during the early stages of the development of a *change*, where many of implementation details are unknown.

Assumptions may include:

- the context in which the *change* will be operated – this is particularly critical when developing a new product, in order to explore all the *hazards* associated with the product
- areas of the *TAS* which will not be impacted by the *change*

- other *changes* which will be made to the *TAS* before this *change* enters operational service
- technology which will be used

It may also be necessary to assume that the current level of safety within the system is tolerable.

Assumptions are often made because they relate to behaviour or properties in another *domain* of the *TAS*. The concept of modularisation of the *safety argument* (see sections 5.3, 6.4.3.1 and 6.5.1.3) provides support for ensuring that these *assumptions* are recorded and agreed by all parties involved to ensure that they are correctly managed.

In theory, no *assumptions* should be left over at the point of acceptance; however in reality some *assumptions* are made which cannot be validated before *approval* is granted. A register of *assumptions* should be established to record and monitor all *assumptions* made. This register should include the reasons for the *assumption* and the parts of the development which are affected by the *assumption*. The *assumptions* should be evaluated to identify which need to be validated in order to obtain *approval*, and which may remain as *assumptions* when approval is granted. The *assumptions* register must be kept up to date during the development of the *change*.

During the development of the *change*, *assumptions* should be evaluated and where possible validated; it may then be possible to convert them from *assumptions* into statements. Where an *assumption* is found to be incorrect, the impact on the development must be assessed.

It is inevitable that further *assumptions* will be made as the *change* is developed: it is critical that these *assumptions* are fully captured and that their impact on the *approval path* (and the supporting assessment) is fully evaluated (see section 6.8).

Some *assumptions* may remain even at the end of a development. For example, when developing a new product, assumptions will be made about:

- the environment in which it is to be operated
- how it is used
- how it will be maintained.

These assumptions then become *limitations* on how the product must be used in order for its *approval* to remain valid.

6.3.4 Impact of the *Change* on Safety

The impact of the *change* was initially covered in section 4.3.

During the initial definition of the *change* it is important to define the impact of the *change* sufficiently to understand which *domains* of the *TAS* will be affected by the *change*, so that the *approval path* for the *change*

can be developed (see section 6.4).The impact assessment needs to consider both operational systems and support systems[24]. At this stage it is not necessary (and probably not possible) to understand the full details of the effect on the different parts of the *TAS*: however, it is important to understand as fully as possible which parts of the *TAS* will be affected, in order to build as complete a view of the *approval path* as possible. Of course, there may be effects which are not apparent at this stage: when these become apparent the impact evaluation needs to be revised accordingly.

Some *changes* will introduce interfaces between *domains* where these interfaces did not previously exist: these interfaces require close attention to ensure that they are fully captured in *assurance contracts* between the affected *modules* of the argument; it is also necessary to ensure that the affected parties understand the importance of the interface and to ensure that it is fully incorporated into their processes.

As the *change* is assessed, the impact of the *change* on the safety of the *TAS* will be further explored: but this will largely be done through the various safety assessment processes, as described in section 6.5.

When completed (prior to application for *approval* of the change) the *safety argument* will need to establish:

- which parts of the *TAS* are affected by the change
- that each affected part of the *TAS* has been analysed to identify and set safety requirements
- that safety requirements for each affected part of the TAS have been satisfied such that the acceptable level of safety is achieved – see section 6.3.8

6.3.5 Stages of the *Change*

As discussed in section 4.5, complex *changes* are often developed in multiple stages.

When defining a *change* comprising multiple stages, each should be treated as a separate *change*, with its own definition and *approval* path, albeit as part of an integrated overall *change*. This is equally true for *changes* where stages are aligned to different parts of the lifecycle, as for *changes* which are aligned to different operational states of the *TAS*.

The *change* definition should be revisited at the beginning of each stage to ensure that any alterations resulting from previous stages are taken into account and that correct definition of the (stage of the) *change* and the *approval* path (including the individual stages in each case) is maintained.

6.3.6 Transition into Service

For each *change* a process of transition will be followed to introduce the *change* into service. This transition (from pre-change TAS to post-change TAS) needs to be fully defined and assessed to ensure that it can be completed safely. Various aspects need to be considered, including:

[24] Support systems include training systems, test and development systems, contingency facilities, etc.

- preparation for operation
- implementation of arrangements for safety management, change management, configuration control
- planning of the actual switch-over process
- assessment of the switch-over
- definition of reversion arrangements

It is unlikely that all this detail will be available at the beginning of the change lifecycle, and for some changes this transition will be very simple; however the transition does need to be defined and assessed early enough to make sure that all the required arrangements have been made before introducing the *change* to service.

Where a *change* is staged, the transition needs to be defined for each stage.

6.3.7 Organisations Involved

It is important to identify who will be involved in the change, including:

- the change leader (driving the change)
- the *applicant* (who will apply for *approval*)
- the *approver* (responsible for approving the change)
- the *TESG* (co-ordinating the engineering and safety aspects of a major / complex *change*)
- any other organisations affected by the *change*

Roles and responsibilities within the ASCOS Method, including identification of who will be involved at the various stages of the *change*, are discussed further in section 7.

Identification of the organisations and *domains* involved in the *change* will also help to determine the *modules* of the *safety argument*, as these are usually aligned such that interfaces between *modules* correspond to interfaces between organisations and / or *domains*, using *assurance contracts* to capture the dependencies between organisations which need to be fulfilled in order to make the *safety argument*.

6.3.8 Acceptable Level of Safety

When a *change* is made to the *TAS*, it is necessary to determine the level of safety which the change needs to achieve – this is the *acceptable level of safety*. Changes are made for many reasons and often have no intention to improve the level of safety: for such changes it is usually acceptable to demonstrate that the existing level of safety is maintained.

The level of safety must be considered across the whole *TAS*. It is conceivable that a change may improve safety in one *domain* while having a negative impact (i.e. worsen safety) in another *domain*. It is often difficult to justify such a change. To do this, it would be necessary to provide a robust quantification which demonstrates a significant overall positive impact on safety. Production of such a robust quantification is made more difficult by the fact that different *domains* use different types of targets (often with different units), making it difficult to create valid comparisons between *domains* (see section 6.3.8.1). A corresponding

assessment would be needed in the event of a *change* with differing impacts on different sovereign states. (A recommendation for further research in this area is made in section 8.3.7.)

The effect which the *change* will have on the safety of the *TAS* must be taken fully into account. Any safety assessment must include both:

- the positive effect (usually from the design intent of the *change*) to improve the safety (i.e. decrease overall risk)
- the potential negative effects (usually arising from failure to achieve the design intent or from deviations from it)

Note: it may be acceptable for the *change* to maintain the existing level of safety, especially where the intent of the *change* is not related to making a safety improvement in the *TAS*. However, the impact of the design intent on the *TAS* should still be considered to ensure that there are no unforeseen negative effects.

Probabilistic Risk Assessment models (such as the Safety Risk Model developed in ASCOS WP3 – see D3.6 [5]) can be used to perform an early evaluation of the impact of a proposed *change* on the safety of the system. However, it is important to fully understand the scope of any model to ensure that the full effect of the *change* is considered.

When evaluating an improvement in safety, the evaluation of reduction in risk should include the effect of removal of existing elements of the system which are being replaced. (For example these may be obsolete or difficult to repair, and this may have a knock on effect on the safety of the system when these are in use.)

The actual level of safety deemed to be acceptable may be an absolute level or it may be relative (e.g. that the changed *TAS* should be no less safe than the existing *TAS*). Where a *change* is made in the context of an existing Safety Management System (SMS), the *acceptable level of safety* may be defined in that SMS. For wider-ranging changes, the *acceptable level of safety* should be defined by the *approver*.

Usually the *change* is not replacing the whole system for which the target is defined and any target level of safety therefore needs to be apportioned to allow for the risk contributions from other parts of the TAS; often the level of these contributions needs to be assumed.

For a *change* which takes a (purely) compliance based approach, the level of safety may be expressed in terms of compliance with a set of regulations. However, the *acceptable level of safety* is then implicit in the sense that it is the level of safety achieved by a *change* which complies with the regulations.

6.3.8.1 Difficulty of comparison between *domains*

Although we would like to live in an accident free world, we accept that accidents happen. We attempt to reduce the risk of accidents to the lowest level we can realistically achieve, while accepting that a level of risk is a necessary byproduct of aviation.

Because accidents are (thankfully) rare, and because the sequence of events leading from measurable events to accidents is not always well-understood, we cannot always sensibly set targets on the rates of the accidents themselves. Instead, we set targets on events which can lead to accidents if certain mitigations (often beyond our control) fail. There is significant uncertainty over the propagation between the event against which we set the target and the actual accident.

As a result, we have targets within different domains which are related to precursors to the accidents, rather than the accidents themselves, and which are expressed in different units (because they are expressed in the units which make sense within the domain). The regulations are designed around achieving these targets.

When we make a *change* to the *TAS*, we need to determine the *acceptable level of safety* for that *change*. For *changes* within a single *domain*, we apportion the overall target for the *domain* to derive a target for the part of the *TAS* which we are changing.

Where a *change* spans multiple *domains*, it is necessary to demonstrate that the *change* is acceptably safe in all affected *domains*. Ideally, we would agree a single target for the level of safety to be achieved by the *change*. In order to do this, we need to build a model of the whole *TAS*, to allow us to link all the causes from the different *domains* together and then to derive targets for the particular parts of each *domain* which will be affected by the *change*. The whole industry (*authorities*, *applicants* and other stakeholders) needs to have sufficient confidence in that model to accept the derived targets, and to allow them to be used instead of the accepted targets within each *domain*.

The industry has developed various accident models, including the ASCOS model (see D3.6 [5]). These are useful in evaluating risks and the impact of *changes* (see discussions elsewhere in this report). In the long term it may be possible to use such a model of the whole *TAS* to derive targets tailored for a specific *change*, to be used instead of the current "generic" targets. However, the models are not yet at the level of maturity needed to allow them to be used in this way. Until this level of safety is achieved, changes using the ASCOS Method need to apply the existing approaches and targets within each domain.

6.4 Develop the *Approval Path*

Once the *change* has been defined, the next step is to develop the path[25] to be followed in order to obtain *approval* for the change.

It should be noted that the *applicant* must satisfy themselves that the change is acceptably safe; they may have a legal responsibility to do this under national primary legislation[26]. This need may seem obvious, but it can be lost in the focus on gaining external *approval*.

[25] The concept of an *approval path* was introduced in section 3.2.
[26] For example, the UK Health and Safety at Work Act 1974 [26] sections 2(1) and 3(1).

For most aviation *changes*, the *applicant* must then also demonstrate the safety of the *change* to the satisfaction of the relevant *approver* before the *change* is brought into service. If the *applicant* cannot satisfy themselves then it is very unlikely that they will be able to convince the *approver*.

This section provides guidance on the process of developing the *approval path*:

- defining and developing the *approval path* itself
- developing the *modules* of the *safety argument* needed to support the application for *approval*
- identifying who needs to be involved in the process
- developing and agreeing the *approval plan*

In addition, some guidance is provided on:

- how the ASCOS Method might be different when making organisational *changes*
- the role of standards in *approval* submissions

As explained in section 3, the concept of an *approval* path encompasses the current concept of establishing a *certification basis* and certification plan, but deliberately widens the scope to also include *changes* where *certification* is either only a component of the *change* or does not feature at all.

6.4.1 Define the Top Level *Safety Argument*

The first step towards developing the *approval* path is to define what claim(s) is / are being made about the *change* in order to support the application for *approval*. It is critical to ensure that the *approval path* focuses on demonstrating the correct *claim*; otherwise it is easy to waste effort on activities which are included in standards but not actually required to support the *claim*.

A generic top level claim is presented in section 5.2.2: “Change X to the TAS is acceptably safe”. Definition of the change was handled in section 6.3. The concept of an *acceptable level of safety* is handled in section 6.3.8. Note: “acceptably safe” may indicate maintaining the current level of safety.In developing the *safety argument*, it is necessary to ensure that the whole lifecycle of the change is considered. This is the reason that the generic argument (repeated from section 5) is decomposed into five sub-claims to demonstrate that:

- the actual changed part(s) of the *TAS* are (predicted to be) safe in operation (claims 1-3);
- the process of introducing the change is safe (claim 4);
- the safety of the changed *TAS* in operation will be monitored to check whether the *acceptable level of safety* is achieved, and to address any deficiencies found (claim 5).

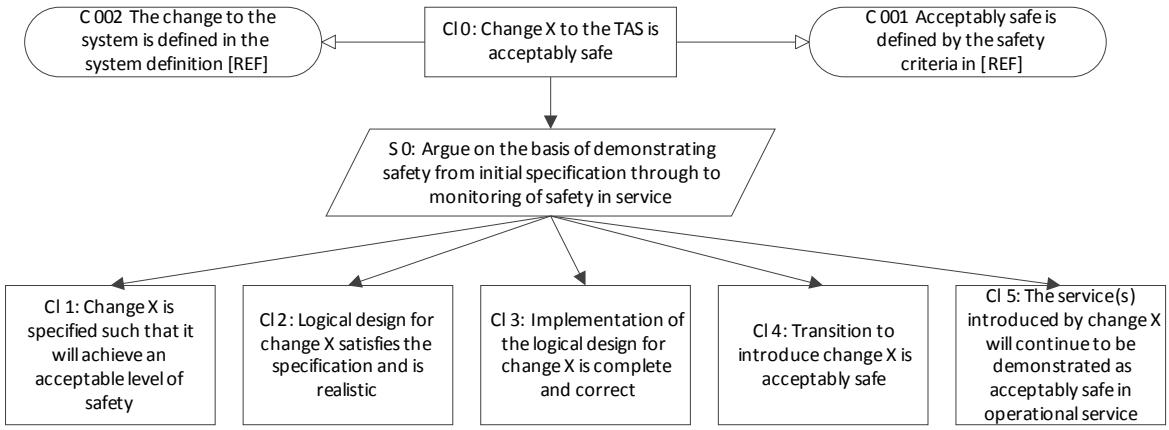

Figure 20: Generic Logical Argument

Although it is not mandatory to present an argument comprising these five *claims*, they provide a helpful structure to ensure that all these aspects are considered. At this stage each of these *claims* should be reviewed and, if necessary, adapted to the requirements of the specific *change*.

As explained in section 4.5.1, not all *changes* lead directly to an alteration in the operation of the *TAS*. For example, the development of specifications for an RPAS to be operated in unsegregated airspace may be considered as a *change* in its own right. It is still (obviously) important to ensure that the specifications developed specify a (conceptual) RPAS which would achieve the *acceptable level of safety* if implemented, and the ASCOS Method provides a framework for doing this. This (narrowly defined) *change* would focus on claim 1 of the *safety argument* and will not be able to demonstrate that claims 2-5 are met. However it is still useful to consider the specification against these *claims*, and to consider whether anything in the specification developed would make it difficult for these *claims* to be made. This will facilitate the later development of the full *safety argument* for introduction into operational service of an RPAS developed to these specifications.

Where a *change* consists of multiple stages (see section 4.5), the *safety argument* must be demonstrated for each stage. Depending on the structure and size of each stage, it may be appropriate to develop a separate *safety argument* for each stage.

It is also necessary to partition the *safety argument* into *modules* representing the different *domains* and organisations involved and to establish *assurance contracts* between the *modules*, as a means of managing[27] the dependencies between the *modules*. The owner of each *module* will need to demonstrate that the top level *safety argument* is satisfied to the extent of the owner's responsibility for the safety of the *change*. It is useful at this stage to develop an outline *module* diagram showing the interactions between different parts of the *TAS*. The intention of this diagram is not to define all the (many) functional interfaces; instead (especially at this stage) it serves to identify the *assurance contracts* which need to be established in order to support the *safety argument*. In the example diagram shown in Figure 21, the *modules* are shown in boxes and the lines between them represent individual *assurance contracts*.

[27] The importance of managing these interfaces is discussed in section 5.3.2.

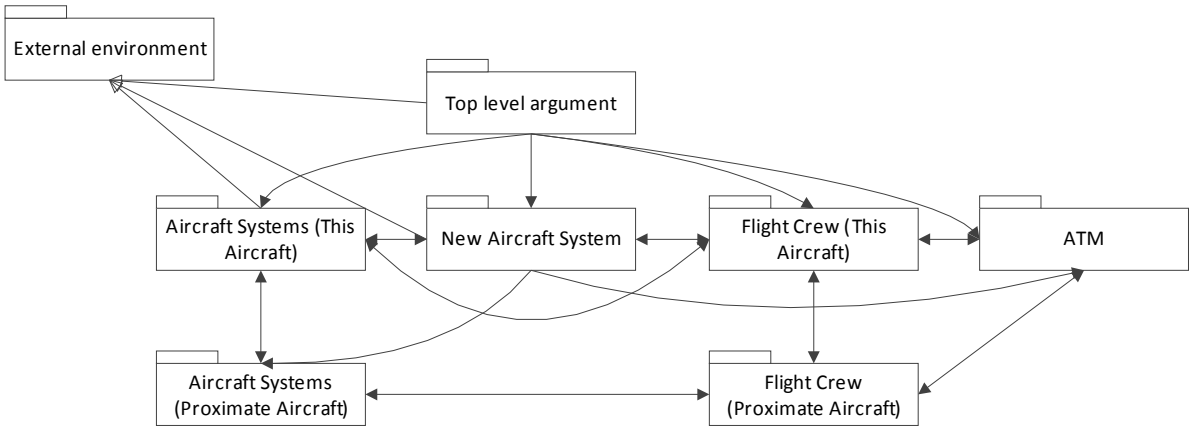

Figure 21: Example of modularisation of argument

6.4.2 Evaluate the Existing Approval Path(s)

Once the overall *safety argument* has been defined, it is necessary to develop the *approval* path which will be taken to support the *safety argument*. The first step of developing the *approval path* is to review existing approaches used within the industry and identify:

- how existing approaches can be applied to support the *safety argument*;
- where there are gaps in existing approaches which need to be filled in order to fully support the *safety argument*.

It is helpful to review the implicit arguments[28] already used for *approvals* within the TAS as these provide insight into how the existing approaches support a *safety argument*.

When evaluating existing *approval* paths, the following questions should be considered:

- **Does the existing path address all the claims made by the top level *safety argument*?** For example, does it fully ensure that the change will be ready to enter operation? Does it define the monitoring to be undertaken after entry into operation to ensure that the *acceptable level of safety* is achieved?
- **Can the existing *approval* path be made more efficient, while still addressing all the claims of the top level argument?**
- **Does the existing *approval* path fully balance the safety improvements made against the additional risks introduced by the *change*?** For example, the existing approach may be biased towards considering only the failures of the new (part of the) system by deriving a failure rate and comparing it against a target, without taking into account the safety improvements achieved by introducing the *change*.

[28] It is a recommendation of this document (section 8.3.2) that the arguments implicit within existing approval approaches should be documented to support easier development of approval paths using the ASCOS Method.

- **What assumptions are made by the existing *approval* paths or standards[29]?** There are often significant implicit *assumptions* within existing standards or approaches, including:
    - the type of solution (e.g. based around electromechanical rather than electronic)
    - the environment (e.g. weather conditions, behaviour of wake vortices)
    - the means of operation or maintenance (e.g. piloted operation)

  Are these assumptions valid for the *change*? If not, how does this affect the validity of the approach?
- **Does the *approval* path fully manage interfaces between different parts of the *TAS*?** Does it ensure that any *assurance contracts* between different parts are fully defined and arrangements put in place for maintaining them through the lifetime of the *change*. For example, introduction of a new (type of) organisation within the *TAS* will introduce new assurance requirements to be fulfilled by that organisation.
- **Where the *change* follows multiple *approval* paths (i.e. where the system needs multiple *approvals*, potentially by multiple *approvers*), are the interfaces between these *approval* paths fully managed to ensure that the *claims* made are consistent between the different parts of the *TAS*?**

6.4.3 Develop the *Approval Path*

The results of the evaluation described in section 6.4.2 should allow an initial development of the top level *safety argument*, showing where evidence will be available from existing approaches, and where there are gaps such that further development of the *approval path* is necessary.

For simple changes, such as introduction of a new replacement part in an existing system, there may be no gaps in the *approval path* and it may be straightforward to demonstrate that the overall *safety argument* is satisfied by existing processes. In such cases no further work is needed in designing the *approval path*, and the next step is to identify the stakeholders in the *change* (section 6.4.4).

In practice, the development of the *approval path* for most complex *changes* will involve developing separate *approval paths* for the individual *domains* of the *TAS*, with each *domain* supported by its own *module(s)* of the *safety argument*. Each of these *modules* will make the same essential *safety argument*, but within the *context* of its own *domain* of the *TAS*. Within each *module*, some parts of the *safety argument* will be met by the existing approaches and other parts will need additional approaches to be developed.

It is essential that the *assurance contracts* defining the dependencies between *modules* are fully and correctly defined and agreed between all parties concerned.

The following sections give guidance on developing the argument to the level required to support an application for *approval* of the *change*. The activities described will not necessarily be undertaken strictly in

[29] The role of standards within the approval process is further explored in section 6.4.6.2.

the order in which they are presented here; instead they should be considered in parallel, with the overall goal of developing an *approval* path for the *change*.

6.4.3.1 Modularising the *Safety Argument*

The general principles of modularisation of *safety arguments* were presented in section 5.3. Modularisation allows the overall *safety argument* to be subdivided into *modules*, with formally defined *assurance contracts* between them.

At this stage it is useful to separate the *safety argument* into *modules* whose boundaries are aligned to the responsibilities of the various *domains* within the *TAS*. This is especially true where *approval*s in multiple *domains* and / or from multiple *approvers* are required. (See section 6.4.5 below.)

This has the advantages of:

- making the overall *safety argument* easier to visualise and understand
- allowing *modules* to be developed separately from one another in confidence that the final result will be consistent and correct
- partitioning the *safety argument* such that each *approver* needs only:
    - to consider specified *modules* of the *safety argument*
    - to be assured that the *assurance contracts* at the boundary of those *modules* are correctly implemented

Separation into *modules* can then be used to allow both *applicants* and *approvers* in the individual *domains* to focus on the part of the *safety argument* which is pertinent to their *domain*, while also understanding the relationship between their *domain* and the other *domains*.

Assurance contracts should be established between *modules*, as described in section 5.3.2. Particular care is needed to ensure that the importance of the *assurance contract* is understood by all parties involved, especially where the *assurance contract* introduces an interface which is not currently present within the *TAS*. Consideration should also be given to how the *approver* (who will be responsible for approving the *module* pertinent to the *domains* for which they have responsibility) will be assured that the *assurance contracts* with other *domains* have been adequately satisfied.

Modules can also be used for other 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 *safety argument*;
- as a container for issues relating to integration within the overall *TAS*;
- as an aid to developing the safety requirements for individual parts of the solution, by containing the *safety argument* relating to different products in different safety case modules;

- to separate direct evidence from backing evidence[30] - 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.

6.4.3.2 Decomposing the *Safety Argument*

The way in which the *safety argument* is decomposed will be dependent on many factors including:

- the type of *change*
- the existing approaches available
- the *domains* involved

Decomposition should generally be guided by two principles:

1. Does the combination of sub-*claims*, when taken together, prove that the parent *claim* is true ?
2. Have the sub-*claims* been formulated such that the *claim* can be supported by evidence?

In addition, decomposition is used here to distinguish between parts of the *safety argument* which

- are supported by existing processes
- need additional processes to be developed and applied

in order to support the higher level *claim*. Some ways to develop these additional processes are covered in section 6.4.3.3.

This section gives some options for decomposition of the *safety argument*, building on the general guidance given in section 5.2.3. A combination of *strategies* may be necessary and the choice of decomposition should be carefully considered as it has a significant impact on the ease with which the *safety argument* can be managed. (It is assumed here that the *safety argument* has already been split into *modules* to align with *domains* according to the modularisation principles discussed in section 6.4.3.1.)

- **Decomposition by stage**: if the *change* is subdivided into multiple stages which have significant differences (i.e. they are not just progressive deployments of identical technology), the *safety argument* might be decomposed at the top level to form distinct arguments for each stage, before introducing (for each stage) claims 1 – 5 of the generic argument.
- **Decomposition by process**: if multiple processes can be combined to support a *claim*, sub-*claims* might be generated for each process involved.
- **Decomposition by subsystem**: if different subsystems are addressed by different sets of processes with different supporting evidence, it is sometimes appropriate to provide a separate sub-claim for each subsystem.

[30] See section 5.1.1

In each case, it is important to ensure that the sub-*claims*, when taken together, fully support the parent *claim*. The *safety argument* should be supported by a narrative explaining the decomposition and justifying that the entire parent claim is supported. In the GSN notation, symbols can be used to indicate such justifications in the diagrammatic presentation of the argument.

It is easy to make the mistake of decomposing the *safety argument* too far; the *safety argument* should only be decomposed as far as necessary to identify the specific processes and evidence which will be needed to support the top level *claim*. Other pitfalls to be avoided in the development of *safety arguments* are covered in section 5.5.

6.4.3.3 Addressing Gaps in the *Safety Argument*

Once the *safety argument* has been decomposed as described in section 6.4.3.2, any gaps in the *safety argument* need to be filled. These gaps will arise where existing approaches or specifications do not provide all the evidence needed to support the *safety argument*. This part of the process is likely to be iterative, with further decomposition required where a particular technique does not provide all the evidence required to support an individual *claim*.

What is needed to fill the gaps will depend on the nature of the gap. This will range from minor adaptation of existing approaches through to development of completely new means of assessment. As a guide, it is likely that major *changes* to the *TAS* (e.g. introduction of self-assured separation) will require more extensive development of new approaches, whereas smaller *changes* (e.g. introduction of a new feature within an existing aircraft) should be achievable through adaptation of existing approaches.

As development of the *safety argument* is a creative process it is not possible to give a prescriptive guide of how to fill gaps in every case. Instead, Table 4 presents some of the gaps which may be found in existing approaches and gives some guidance on how to go about filling these gaps.

Potential Gap	*Filling the Gap*	*Example*
Existing approach focuses on detailed assessment of designs and does not consider changes at the level of the *TAS*.	Adopt concept level approach for initial assessment. This could be developed from the scenario-based approach developed and successfully applied by EUROCONTROL (see “Safety Assessment Made Easier” [27]).	Introduction of self-assured separation affects the underlying principles of operation for all domains of the TAS and will therefore require comprehensive assessment at the TAS level.

Potential Gap	*Filling the Gap*	*Example*
Change introduces a new interface between domains, or significantly alters an existing interface.	Identify who will be responsible for the interface within both domains. Adapt the assessment approach(es) to include this interface and establish dependencies in both directions across the interface. Formalise these dependencies in an *assurance contract*.	Shift of responsibility for ground de-icing to a new organisation could introduce a new interface where flight crew depends on (staff within) the de-icing organisation to assure that the plane is ice free.
Assumptions made by existing approaches no longer valid (this may include *assumptions* about the environment in which the *change* will operate).	Undertake an impact assessment of the deviations as a consequence of the *assumptions* made for the *change*. Where necessary, adapt the approaches accordingly.	Existing certification specification assumes that presence of a pilot in the cockpit provides mitigation for a number of hazardous occurrences; RPAS no longer has pilot in cockpit and therefore needs to introduce alternative mitigations.
Change beyond the scope of existing specifications.	Undertake a gap analysis of the existing specifications and develop specifications (or performance based requirements) to address the gaps.	The certification specification for light rotorcraft (CS-27 [28]) does not cater for RPAS; JARUS has developed the (CS-LURS [29]) to extend the scope to RPAS of this type.
Existing approach does not balance safety benefit against risk of failure.	Adopt an approach which fully considers the safety benefit. This can be done by evaluating the *inherent hazards* within the *TAS* which the *change* is intended to mitigate, in order to understand the benefit gained implementing the *change*; this must then be offset against the disbenefit from the potential failures introduced by the *change*. Note: in practice it is very difficult to construct an argument to support any increase in risk, even where this is offset by a significant decrease in risk elsewhere – see section 6.3.8.	Where introducing an automated aircraft recovery system (AARS), the hazards resulting from the operation (or failure) of the AARS would be identified by “classic” hazard assessment techniques. However, if the advantage (and raison d’être) of the AARS in preventing crashes is not taken into account, introduction of an AARS could easily appear to be compromising safety.

Potential Gap	*Filling the Gap*	*Example*
Existing approach does not justify the safety for the individual stages of the *change*[31].	Where the stages are small increments culminating in the overall final *change* it may be sufficient to undertake the main assessment on the final *change* and then undertake smaller assessments to look at the differences between the final *change* and the initial stages. Where each stage represents a *change* in its own right, it may be necessary to undertake a separate full assessment of each stage as a *change*.	Change in surveillance technology to be introduced progressively across an area of airspace in multiple stages. Impact during interim stages of having multiple technologies in use needs to be assessed.
Existing approach does not consider all the stakeholders who will interface with the changed (part of the) *TAS*.	Ensure all stakeholders are involved in the assessments undertaken at the *TAS* level; during these assessments scope the further involvement needed at more detailed level.	Development of airborne / cockpit equipment does not always fully consider the practicality of operating or maintaining the equipment.
Existing approach does not fully consider the impact of the *change* on all parts of the *TAS*.	Extend existing approach to consider the impact on all parts of the *TAS*.	The example of the AARS (see above) is also applicable here.
Multiple *approval paths* not integrated.	Define assumptions, scope and context for each *approval* path; review these against each other and address inconsistencies where these occur.	Development of an RPAS will require (inter alia) type certification of the aircraft, *approval* of new (ATM) operating procedures and modification of pilot training requirements. These will all need *approval* by different *approvers*, but there is no automatic process to show how the development in all these *domains* remains consistent. The *approval plan* would need to show how interfaces between these *domains* are fully managed.

Table 4: Gaps which may arise in the approval approach

[31] This envisages *changes* subdivided into stages representing different operational states of the final system – i.e. where a *change* is introduced into operation incrementally.

6.4.3.4 Improving Existing Processes

Consideration should be given to improving existing processes, even where no specific gaps are identified. Whether the benefit justifies the cost depends on the scale of the *change*, and the degree of improvement made to the processes should be tailored accordingly.

For major *changes* spanning multiple *domains* of the *TAS*, it may be worth considering development of a harmonised framework of development and assessment processes for the *change* to streamline the processes and make them consistent across the development of the *change*. Development of such a harmonised framework is recommended by ASCOS WP3 (see D3.6 [5]). In the longer term, it is envisaged that this harmonised framework will be captured in standards applicable across the aviation industry, but this framework would not be available for early implementation of the ASCOS Method.

The proposal from WP3 includes a standards hierarchy which is harmonised across the domains. It also introduces feedback loops so that where faults or failures are traced to shortcomings in the processes, the processes involved can be updated accordingly to address these shortcomings.

Another source of improvement is from lessons learned by other changes to the *TAS*. Such lessons are often not readily shared between organisations – the reasons for this include lack of funding for the effort involved and concern over releasing commercially sensitive information[32]. However, the *argument architect* should make use of any information which can be gleaned to streamline the processes adopted.

6.4.4 Determine Stakeholder Involvement

Every *change* will have a *change leader*, the organisation which is driving the *change*. *Changes* will usually also have other stakeholders. All stakeholders must be identified so that they can be fully involved in the process.

Stakeholders will usually include:

- the *applicant* who is requesting *approval* for the change (often this will be the same organisation as the *change leader*)
- the *approver* responsible for approving the *change*
- stakeholders affected by the *change*, but not directly involved in making it (for example, introduction of an RPAS will require changes to ATM practice and to pilot procedures, and representatives of these domains should be consulted.)
- stakeholders providing a product or service which forms part of the changed *TAS* (for example equipment manufacturers or telecoms service providers)

It is also important to identify the *argument architect* who will have responsibility for the developing and maintaining the *safety argument*.

[32] See the recommendation in section 8.3.3.

It is necessary to identify, for each stakeholder:

- the stakeholder's role in the *change*
- the stages of the *change* in which the stakeholder is involved

For complex *changes*, a stakeholder's role may vary during the lifecycle of the *change*, as described in section 4.5.1.

It may be necessary to make multiple applications for *approval*, and for multiple approvers to be involved, especially where the *change* spans multiple domains of the TAS. Such *changes* will involve multiple *applicants* and *approvers*.

Roles and responsibilities within the ASCOS Method are discussed further in section 7.

6.4.5 Plan for *Approval*

Many changes will need formal *approval* before they are brought into service. Development of the *approval path* and *safety argument modules* should be followed by development of an *approval plan* which shows how the *safety argument* will be presented to the relevant *approver* for *approval*, including the supporting evidence which will be presented. Often, multiple *approvers* will be involved; in these cases an overarching plan should be developed and submitted to all *approvers* involved to show how the individual *approvals* are related – this may need to be supported by further *approval plans* presenting the details relevant to individual *applicants* and *approvers*.

Subdivision of the *safety argument* into *modules* should simplify the *approval plan* as it should be possible identify a single *approver* for each module; although it will also be necessary to demonstrate how the *approver* will be assured that the *assurance contracts* between that *module* and the rest of the *safety argument* will be satisfied.

The intention of the *approval plan* is to explain how the *applicant* intends to demonstrate that the change achieves the *acceptable level of safety*, including the evidence which the *applicant* will present to support the change. Based on the information provided in the *approval plan*, the *approver* will undertake their own assessment of the change and determine the level of involvement which they will have in reviewing the change.

The *approver* may define a specific process to be followed in order to gain *approval*. The *approval plan* should show how the *approval* path developed by the *applicant* is aligned to that process.

The *approver* will not give formal *approval* at this stage, but early involvement:

- gives the *approver* early visibility of the proposed change
- enables the *approver* to explain their requirements, which may include:

- form of argument expected
- type of evidence expected
- time and resources required by the *approver* to review the submissions

- enables the *applicant* to adapt their argument and supporting approaches to address the needs of the *approver* before significant effort has been invested in generating inappropriate evidence

It is very strongly recommended that the *safety argument* and the proposed supporting evidence should be agreed between *applicant* and *approver* at this stage. If this agreement is not achieved at the start of the development, there is a significant risk that the *safety argument* and evidence produced by the *applicant* will not be acceptable to the *approver*. The *applicant* may then need to incur significant extra effort (and significant delay) in order to produce the evidence required. At worst, the *approver* may be completely unable to accept the proposed change.

For complex or wide-ranging changes, it may not be possible to demonstrate that the product or concept is (sufficiently) safe across the whole desired range of operation: in this case initial *approvals* may be restricted to cover only the range of operations which have been demonstrated to be safe. Where possible, this staged introduction should be planned into the deployment and covered in the *approval plan*.

The *approval plan* should include the following elements:

1. **An overall description of the change**

   An overall description of the *change* for which *approval* will be sought, its limits and the way it is interfaced with other *domains*. This description is primarily intended for the experts of the *approver*. It should highlight relevant aspects such as technical novelties and, where appropriate, relationship with other *domains*.

2. **The *approval path***

   A presentation of the *approval path* to be followed, including the supporting *safety argument*[33], along with a clear indication of the *modules* of the *safety argument* which each *approver* is expected to approve.

3. **Management of requirements**

   The *approval plan* must list the applicable regulatory requirements (for a *certification* this would be the certification basis) and related guidance material. It should also put in place a framework for resolution of any issues with the requirements: issues may arise either because aspects of the development are not covered by requirements, or because the development conflicts with existing requirements. The rationale for any such deviations should be underpinned by the *safety argument*.

[33] Where the argument is presented in graphical form (e.g. GSN) there should also be a narrative which explains how the argument is structured.

4. **Planned evidence**

   A list of the evidence which is proposed to support the *safety argument* (for a *certification* this would be a means of compliance checklist), and which parts of the evidence will be presented to the *approver*. This will include evidence to support all parts of the *safety argument*, including the arrangements for transition into operation (claim 4) and for ongoing monitoring of the *change* while in operation (claim 5).

5. **Programme for production of evidence**

   This programme ensures that all stakeholders agree over when the evidence is to be produced, taking into account the constraints imposed by the development and validation of the system as well as the *approver's* review timescales. This programme may be incorporated into the list of evidence to be produced.

6.4.6 Supporting Information

The previous sections presented the steps in development of the *approval path*. (As discussed in those sections, development of the *approval* path is not a single linear pass through these steps and will require a degree of iteration.)

The following sections present supporting information addressing specific issues which may arise in the development of the *approval* path. In particular, they address:

- Section 6.4.6.1 - how the ASCOS Method described may need to be varied where the change being made is primarily an organisational change, rather than a change to the technical systems within the TAS.
- Section 6.4.6.2 – how standards can help (and hinder) the development of an *approval* path.

6.4.6.1 Organisational Change

Some *changes* to the *TAS* are changes to organisation rather than changes to equipment or processes. For example a *change* may introduce a new type of licensed organisation, or the *change* may be to license a new organisation. The *change* may be introducing a new function within the *TAS*, or it may be transferring responsibility for an existing function to a new organisation.

The overall focus is still on the *change* being made to the *TAS* and how this will affect the safety of the *TAS*.

The *safety argument* still needs to address fundamentally the same questions; however, there will be differences in

- the way in which the *change* is defined
- the structure used to decompose the *claims*

- the means of analysis
- the process followed by the *approver* to evaluate the argument and grant *approval*

In defining the *change* it will be necessary to define at a functional and performance level what service is being provided by the organisation, and how the existing service provision (if any) is being modified by the *change*. (For example, licensing a new air operator may introduce new services (of the same type) into the *TAS*; this may have an impact on existing services by putting pressure on availability of stands or runway slots for existing operators.)

It should be noted that, depending on the service provided by the organisation, the focus may be on safely delivering a particular level of service to support other organisations, rather than on directly delivering a particular level of safety. This is the case where an organisation does not have a direct effect on the safety of the system.

The top level *safety argument* can be arranged around the same basic *claims*; the following list provides guidance for how each could be addressed:

- **does the *change*, as specified, achieve the *acceptable level of safety*? (claim 1)** The scope of the functions delivered by an organisation, and the *acceptable level of safety* (service) to be achieved needs to be specified in regulations. In support of claim 1, the organisation using the ASCOS Method to apply for a licence may need to do nothing more than cite the applicable regulations and explain why they form a complete specification of their operation. In this case their *safety argument* may have a built-in assumption that the specification has been designed to deliver a function to an *acceptable level of safety*. A separate application of the ASCOS Method may be used by the *authority* responsible for the specification to demonstrate that this is the case.
- **does the *change* as designed (claim 2) and implemented (claim 3) achieve the *acceptable level of safety*?** It will be necessary to demonstrate that the organisation can deliver the *acceptable level of safety* (service) and that any possible failure modes in the service provision have been identified and suitable mitigations put in place as necessary.
- **will the transition to the new arrangements be managed safely? (claim 4)** Where provision of an existing service is being transferred to a different organisation, how will this transition be managed to ensure that safety is maintained during the transition? Where a new service provider is being licensed (e.g. a new air operator) how will this affect existing operations? (For example, how will any new arrangements be briefed to ground staff so that they know how to accommodate a new air operator within the aerodrome's operations?)
- **how will safety be monitored following the transition? (claim 5)** This should be through a combination of the new organisation's SMS and through the monitoring of the organisation by the authority responsible for oversight of its operation.
- **how are the organisation's interfaces with the rest of the TAS established and managed?** An assessment of the organisation's interfaces with other organisations will be needed. These interfaces

will need to be formally defined and assessed, to ensure that any dependencies between the organisations are fully understood and captured.

Note: the ASCOS Method does not, in itself, prescribe the specific techniques to be followed. As noted previously, WP3 of the ASCOS programme has proposed an approach to harmonisation and improvement of the standards used for safety assessment across the domains of the TAS - see ASCOS WP3 Final Report [5], section 6.3.

6.4.6.2 Role of Standards

Standards are part of a hierarchical regulatory framework which may be viewed broadly in three tiers:

1. regulation and legislation
2. guidance on compliance with regulation
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.

The first tier (regulation and legislation) is the group with which it is mandatory to comply.

The second tier provides guidance, from a variety of sources, on how to comply with the regulation and legislation in the first tier. This includes Acceptable Means of Compliance (AMC) published by EASA, and Alternative Means of Compliance (AltMoC) and other guidance published by other competent authorities.

Material in the third tier includes:

- EUROCAE documents such as: ED-79A / ARP 4754A [30], ED-109A [31], ED-125 [32]
- industry standards such as DO-178C [33] and IEC 61508 [34]

The contribution of guidance (i.e. tiers 2 and 3) to the *safety argument* for *approval* is mixed. It can be viewed as an enabler, a constraint or as a tool to provide consistency of approach.

It should also be noted that little of the guidance available directly addresses safety: it is largely focussed on best practice and interoperability.

Application of a standard can provide a clear and concise set of evidence to support the *safety argument* for a change. The advantage of correct application of established standards is that they can be used to generate a set of evidence which is readily understood and readily applicable to multiple developments. In addition, less training and familiarisation is required, meaning that the evidence can be more readily produced.

However, use of standards can present a number of pitfalls, which should be guarded against. The main pitfalls relate to:

a. the underlying *safety argument* assumed by the standard

b. the context within which the standard is applied

If the underlying *safety argument* assumed by the standard is different from the *safety argument* developed for the change, then the evidence generated will not directly match the evidence required for the *change*. It is necessary to evaluate the set of evidence to be generated, to confirm whether it will support the *safety argument* and what gaps will be left; this is especially important where the underlying *safety argument* is implicit and therefore cannot be directly compared against the *safety argument* developed for the *change*[34].

If the standard is not applied within its intended context, the evidence produced may not be usable, because it may make invalid assumptions about the rest of the change. The importance of context is illustrated by DO-178C [33] and DO-160 [35] which are low level, component related standards, providing only a small part of the overall regulations applying to the aircraft equipment. Higher level standards are used to determine how these low level standards are used – attempting to apply them out of context may produce evidence which is simply unusable.

Another example of incorrect *context* can arise when assurance levels are used to drive requirements outside their intended scope. Assurance levels are commonly used to index the degree of rigour required in producing the assurance evidence. This is useful but can fail when the level drives criteria that are not appropriate for the safety argument being made. Most commonly assurance levels are used to drive assurance criteria for reliability of the system, in order to deliver a specified level of risk. Where the argument relates to other properties of the system (e.g. timing, accuracy, robustness, predictability), the same assurance levels may not deliver the required result.

In addition, some standards (e.g. IEC 61508 [34]) are really "meta standards", which require instantiation before they are applied. The instantiation process will involve its own *assumptions* about the context within which the instantiated standard will be applied. Where these *assumptions* do not hold for the *change* being made, the evidence generated may not be able to support the *safety argument*.

6.5 Develop Solution

This section focuses on further development of the *safety argument modules* (including the supporting evidence) in parallel with the development of the *change* itself: this is an extension of the initial version of the *safety argument*, which was developed to support the *approval path*, as described in section 6.4. The aim of this further development is to ensure that, when the application for *approval* is made, it is supported by a complete, correct and consistent *safety argument* including an appropriate body of evidence.

[34] It is a recommendation of this document (section 8.3.2) that the *safety arguments* implicit within existing approval approaches should be documented to support easier development of *approval paths* using the ASCOS Method.

The ASCOS Method does not attempt to replace existing established techniques, either for the development or the assessment of the change, where these are able to generate the evidence needed to support the *safety argument*; however the ASCOS Method does provide guidance on adaptation of existing techniques to ensure that the *safety argument* is complete and fully supported by evidence.

The development of the *safety argument* follows the general work flow shown in Figure 22, which is further explained in section 6.5.1.

When developing the *safety argument*, it is important to bear in mind the following key points.

- **System development lifecycle** – The system will be developed according to its own defined lifecycle; this will be different from the workflow of the ASCOS Method as presented in section 3. It is important to align development of the *safety argument* to the system development lifecycle and to build in appropriate check points as discussed in section 6.5.1.5.
- **Development is iterative** – see section 6.8.
- **Maintenance of the *safety argument*** – see section 6.8.

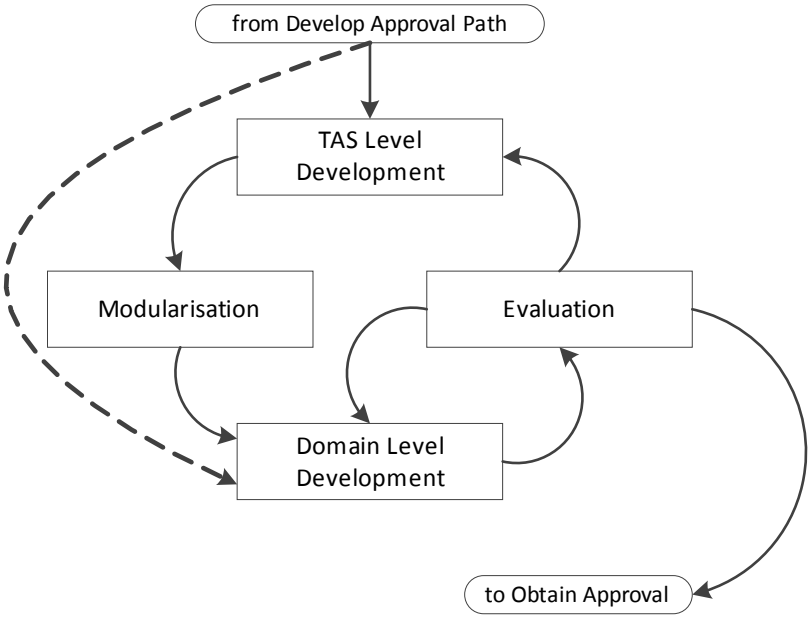

Figure 22: Iterative workflow of argument development

As the solution is developed, the *safety argument* (see Figure 23) will generally be developed from left to right (i.e. from Claim 1 to Claim 5), although a few exceptions are described below. The following sections address the development of the *safety argument* from two different perspectives. Section 6.5.1 (and subsections) considers the different stages of work flow presented in Figure 22. Section 6.5.2 (and subsections) provides further guidance on development of the individual *claims* of the *safety argument*.

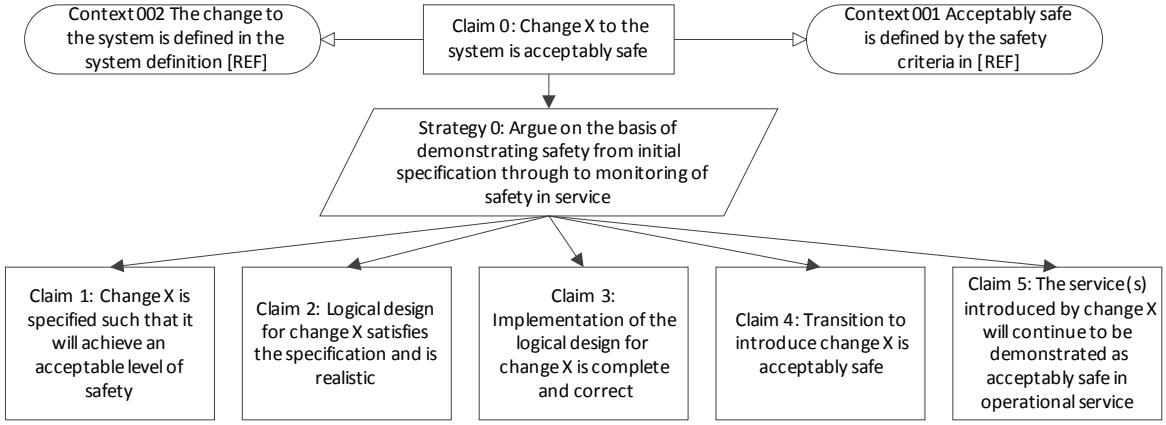

Figure 23: Generic Logical Argument (repeat of Figure 14)

There is a potential for overlap (or at least a moveable boundary) between section 5.4 and section 5.5, because both relate to development of the *safety argument* and it is difficult to define where development of the *approval path* ends and development of the solution begins. The important point is to ensure that the *safety argument* is developed and maintained in parallel with the solution and the evidence required to support the *safety argument* fully is identified and (eventually) generated.

6.5.1 *Safety Argument* Development Workflow

The following sections describe the activities represented by the workflow in Figure 22, followed by observations relating to iteration and the maintenance of the argument.

6.5.1.1 Entry Point – From Approval Path

The entry point into the cycle shown in Figure 22 will depend on the nature of the *change*.

Changes which are focussed on a single *domain* and which have limited impact on the rest of the *TAS* may already have a fully developed *safety argument* at the *TAS* level (see the activities described in section 6.4). Development for these *changes* may follow the dotted line shown in Figure 22 and proceed directly to domain level development.

Other *changes* will require significant assessment at the *TAS* level to take account the overall impact of the *change* on the safety of the *TAS*: these follow the solid line to TAS level development.

6.5.1.2 TAS Level Development

Where a change spans multiple domains of the *TAS*, significant systems engineering and assessment effort is needed at the *TAS* level to ensure that the overall impact of the *change* on the safety of the *TAS* is fully considered. The *safety argument* must be developed in parallel to ensure that it will support the eventual application for *approval*.

Initially the development at the *TAS* level will focus on developing and assessing the specification for the *change* (i.e. supporting claim 1 – see section 6.4.1) and decomposing this into a design (i.e. supporting claim 2) which addresses the *change* within all the *domains* of the *TAS*, with appropriate *assurance contracts* agreed between the *domains*.

Claims 3 to 5 of the *safety argument* also need to be considered at the *TAS* level, but to a lesser extent: the majority of the support for these *claims* comes at the *domain* level, with the *safety argument* at the *TAS* level showing that the evidence at the *domain* level is correctly integrated to form a complete *safety argument* covering the whole *TAS*.

It is critical to ensure that the full impact of the *change* across the *TAS* is considered at this level to ensure that all potential safety effects are identified and assessed. (See sections 4.3 and 6.3.4 on the impact of *change*.) It is also critical to ensure that all relevant stakeholders are involved in the assessments at this level. For example, ATM should be consulted in a *change* which may affect the behaviour of aircraft (e.g. development of an AARS), even if the ATM procedures will not be directly affected.

Section 6.5.2 provides further guidance on developing the argument for each of the top level *claims*.

6.5.1.3 Modularisation

Modularisation of the *safety argument* to align to the *domains* of the *TAS* and the organisations involved has already been considered during the development of the *approval path* (see section 6.4.3.1). However, where significant *TAS* level development of the solution is undertaken as described in section 6.5.1.2, this modularisation should be revisited. This should be done:

- to ensure that the *modules* still represent appropriate subdivision of the solution – it may be necessary to introduce new *modules*, or modify *module* boundaries to reflect the *TAS* level development
- to ensure that the *assurance contracts* still fully capture the dependencies between *modules*, including the context and caveats relevant to the *claims* in the *modules*
- to identify whether additional *modules* should be created to encapsulate details of the *safety argument*.

This modularisation affects all *claims* of the *safety argument* – in each case the *claims* made at the *TAS* level will be decomposed into *claims* within the individual *domains*, with agreed *assurance contracts* between them. (See section 6.5.2 for further guidance on each of the *claims*.)

The initial modularisation (especially of Claim 2) will be in parallel with the systems engineering functional decomposition of the solution into *domains*.

A primary use of modularisation is to separate the *safety argument* into *domains*; however, modularisation can be used for other purposes, as explained in section 5.3.3. The principles remain the same: to subdivide the *safety argument* into *modules* which are easy to develop and maintain as separate units. Care is needed to

clearly identify which *modules* fall into which *domains*, and therefore to identify which *modules* are required to support the *approval* in each *domain*.

6.5.1.4 *Domain* Level Development

Development of the *safety argument* continues within each of the *domains* of the *TAS*.

Initial development of the *safety argument* at *domain* level will focus on supporting claim 2, showing that the design of the *change* is capable of delivering the *acceptable level of safety* in each of the *domains* and that the *assurance contracts* within the *domains* are developed and satisfied. This builds on the initial work at *TAS* level supporting claim 1, which defined the safety requirements on the TAS and the modularisation which apportioned these to *domains*.

Assessment within the *domain* will follow existing techniques where possible (see discussion in sections 6.4.2 and 6.4.3). These may need to be adapted / extended where the *change* introduces concepts not envisaged by existing standards. Common issues with existing techniques were identified in section 6.4.3.3.

The argument supporting claims 3 – 5 will also be developed at *domain* level - see section 6.5.2. Much of this will take place later, once the design is further developed and the solution approaches implementation and deployment. However these claims should still be considered, even in the early stages of the development. In particular, ASCOS D3.5 [5] identifies the importance of early identification of possible precursors (supporting the monitoring required in claim 5), during the safety modelling of the *TAS*, which will usually be conducted in support of claim 2.

It is essential that the evidence needed to support the *safety argument* is clearly stated, and that the assessments take this into account; otherwise there is a risk, especially where practitioners are used to applying the "standard" techniques, that the evidence produced will not support the *safety argument*. (See section 6.5.1.5 on evaluation of the evidence.)

It remains important that all stakeholders are considered throughout the development of the argument. At *domain* level, this is partly addressed through the *assurance contracts* between *domains*. However, the existence of a contractual relationship should only be seen as formalising the requirements: it is no substitute for ongoing engagement with the other stakeholders to ensure that the *assurance contracts* match the needs of the *safety argument* and are correctly understood and accepted on both sides of the interface.

6.5.1.5 Evaluation

At regular intervals, it is necessary to check that the evidence generated by the assessment processes provides the expected support for the *safety argument*, that this support is complete and that the evidence respects the context of the *claim* which it is supporting. It is also necessary to check that the *safety argument* remains appropriate to the *change*. These checks, which should be carried out by the *argument architect*, are necessary because development of *changes* is a creative process and it is possible (or even likely) that the development of the *change* will stray away from what was expected when the *safety argument* was initially constructed. It is

also likely that the assessment will generate *caveats* which need to be addressed further, perhaps through modifying the solution or introducing *limitations* on its application to the *TAS*.

Large programmes are often divided into a number of lifecycle stages[35], with "stage gates" between stages. The programme must be able to demonstrate that certain criteria are met before it can proceed to the next stage of the lifecycle. The stage gates may be an appropriate point at which to evaluate the state of the development and the *safety argument* and to take corrective action as necessary.

Where significant issues are encountered which affect the definition or design of the *change* or the structure of the *safety argument*, evaluation should not be delayed until the next stage gate, but should be undertaken immediately. However, it is important to base the evaluation on a mature and stable understanding of the system and not on a speculative modification which may be proposed. (Although the impact of a speculative modification on the argument may, in itself, be a significant factor, on whether the modification is adopted.)

Table 5 lists some key questions which can be used to perform this evaluation, along with some of the actions which may need to be taken. These questions should be used as a guide – further questions should be introduced as required by the specific argument for the change. Many of these evaluation questions will form part of a good systems engineering process; but they are repeated here due to their critical impact on the development and maintenance of the argument.

Following the evaluation stage, there is a choice of path (see Figure 22) depending on the findings of the evaluation process: where alterations affect the *TAS* level, workflow should return to "TAS level development" (see section 6.5.1.2); other alterations are more local (e.g. modifying the argument within a domain), involving a return to section 6.5.1.4. The workflow described here should be followed (iteratively as necessary – see section 6.8) until the development of the solution is complete, there is a complete *safety argument* supporting the solution and all the evidence required to support the *safety argument* has been produced. The ASCOS Method then proceeds to the "Obtain Approval" step (see section 6.6).

Note: even where a *change* jumps "straight in" to domain level development (see section 6.5.1.1) it may still be necessary to return to *TAS* level development and modularisation, depending on the nature of the alterations required following the evaluation stage.

[35] E.g. concept design, detailed design, implementation, verification.

Evaluation Question	*Answer Requiring Action*	*Corrective Action*
Have the stakeholders' requirements altered since the *change* definition was developed?	Yes	Review the definition of the *change* and update as necessary so that stakeholders' requirements are met. Often it will not be possible to meet all stakeholders' requirements and it is necessary to make decisions about which requirements will be met and which will be "rejected".
Has the definition of the change itself been varied? (For example, due to variation of stakeholder requirements as discussed above.)	Yes	If the change definition is modified, then the *safety argument* will need modification to fully support the new *change* definition. This will include revisiting the modularisation and *assurance contracts* to check whether they are still sufficient to support the new *safety argument*.
Does the evidence produced by the assessments support the claims which are being made? (In the end, will the *change* be acceptably safe, and be demonstrated to be so?)	No	The corrective action depends on the nature of the deficiency. It may be sufficient simply to generate further evidence; if this is not feasible, an alternative *safety argument* may need to be constructed. However if the evidence actually contradicts the *safety argument* the solution may need to be altered.
Is it (still) feasible to produce the evidence called for to support the *safety argument*?[36] (At early stages of development, the evidence will not actually have been produced, but the *argument architect* should still evaluate whether it is feasible to produce the evidence, given the development so far.)	No	Where it becomes apparent that it will not be possible to produce the evidence called for by the argument, an alternative approach (or even an alternative solution) should be sought, depending on the expected deficiency in the evidence.
Does the argument within the *domain* continue to satisfy the requirements placed upon it in *assurance contracts*?	No	The effect on the other *domain* must be considered; the *safety argument* in that *domain* should be modified accordingly and a renegotiated *assurance contract* should be established[37].

[36] This question is related to the previous one, but looks forward to evidence to be produced in the future.

Evaluation Question	*Answer Requiring Action*	*Corrective Action*
Does the *safety argument* within the *domain* rely on other *domains* in ways not already captured in *assurance contracts*? (During development of the solution, detailed assessments will reveal further assumptions about (or requirements on) other *domains*.)	Yes	It is necessary to establish whether the other *domain* can support the *safety argument* in the way required. Explicit renegotiation of the *assurance contract* is needed, to ensure that it is agreed on both sides[37].

Table 5: Evaluation of the development of the argument

6.5.2 Guidance on the Individual *Claims*

All the top level *claims* of the *safety argument* need to be considered, both at the *TAS* level and the *domain* level.

6.5.2.1 Claim 1: *Change* specified to 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 any of the internal details of the *change*. It is important to consider the *change* in terms of high level functions and performance, operational behaviour and modes of operation – including consideration of all the normal, abnormal, degraded and emergency conditions which may occur.

For example, in a change to flight paths into an airport, this would consider the paths which the aircraft take through the airspace, without considering the tasks or equipment employed to guide them to these paths.

It is critical to ensure that the change (in terms of its design intent) is specified to deliver an *acceptable level of safety*, before considering how possible failures may erode that level of safety. Many changes in the aviation system are introduced with the explicit intention of making the system safer: for example, the Automated Aircraft Recovery System (AARS) proposed as one of the ASCOS case studies (see ASCOS D4.5 [36]). It is critical to ensure that the intended improvements are achieved by the *change* and that the *change* does not have an unacceptable (knock-on) effect in other areas of the *TAS*. (Trade-offs between *domains* will not usually be acceptable and would need to be robustly and quantitatively supported – see section 6.3.8).

The assessment at this level focuses on the *inherent hazards*[38] within the *TAS* and the impact of the *change* on these *hazards* – in all *domains*. The assessment should include consideration of the effect of other changes which may be made to the *TAS* during the lifetime of the change being developed. The FAST/EME1.1 methodology (see ASCOS D3.6 [5]) provides a way to evaluate these changes and identify potential hazards

[37] It is easy to make assumptions about another party's activities and proceed without confirming these assumptions. If the other party's activities significantly deviate from the assumption this can leave a significant gap in the argument, which may lie unrectified until a very late stage in the development when it becomes expensive to fix.
[38] These are the *hazards* which exist anyway in the TAS (e.g. CFIT, LOC-I), and are not a result of introducing the *change*.

within the *TAS*; this methodology has already been applied and the results are available [25]. At this level, some consideration of the *introduced hazards*[39] is also possible, but this will be limited to failures at the functional level: assessment of the causes of these failures can only be fully developed when *change* is being assessed at design and implementation levels.

The (change in) risk resulting from the *change* is assessed in order to determine whether this change in risk is acceptable. (This assessment should take into account all hazards which have been identified – both *inherent hazards* and *introduced hazards*, and should be repeated when further hazards are identified.) Where the *change* will not achieve the *acceptable level of safety*, it may be varied (e.g. risk mitigations added) in order to improve safety. If it is not possible to achieve the *acceptable level of safety* through variation of the *change* (including the addition of mitigations) then the *change* must not be implemented.

A possible means of assessment at this level is the scenario based approach described in the EUROCONTROL document "Safety Assessment Made Easier" [27]. The advantage of this approach is that it makes a full consideration of how the *change* will be used within the *TAS* and considers (initially at a high level of abstraction) the impact of the *change* on the *inherent hazards*. Through a variety of techniques this then allows requirements to be developed at a lower level and then flowed out to the individual *domains*. This approach is based on, and thus consistent with, the underlying *safety argument* introduced in section 5.

Whatever means of assessment is used, the important objective here is to develop evidence to support the *claims* that (at the *TAS* level) the change, if it meets the specified requirements, will achieve the *acceptable level of safety*.

The output supporting this claim may include failure models of the system (e.g. ESD, FTA), although causal information will largely be absent because the internal design of the system is not considered in this claim. Such models are usually developed from existing models, for example the ASCOS Safety Risk Model (see D3.6 [5]), derived from the CATS model[40]. Whatever models are used as input, it is critical to understand the scope and context in which they are developed and any limitations implicit in their use. In particular, it is important to consider:

- **completeness** – Does the model represent all the scenarios relevant to the change, across all relevant domains?
- **currency** – Is the model up to date, and does it consider all the envisaged changes which may be made to the TAS during the lifetime of the change under consideration?
- **combination of predictions across domains** – Does the model attempt to compare safety targets between domains? If so, is this approach agreed with all the authorities involved? (See section 6.3.8 for further discussion of this point.)

[39] These are the *hazards* introduced by the *change*.
[40] Development of the CATS model was funded by the Dutch government and led by Delft University of Technology.

As part of the development of Claim 1, the following should be developed and documented:

- the safety objectives for the *change*;
- the safety requirements which specify what the change is required to do (not how it does it) in order to achieve the safety objectives (this should include functional requirements relating to the intent of the change, as well as performance levels which need to be achieved in order to achieve the *acceptable level of safety*);
- the context within which the safety requirements will be delivered;
- the degree of assurance required that the change will meet its requirements;
- any additional functionality requirements or assumptions to capture any external means of mitigating the consequences of hazards
- justification that appropriate processes were used to derive these outputs and that they were applied competently.

The argument supporting claim 1 will be made predominantly at this overall *TAS* level, albeit that the impact within each *domain* of the *TAS* must be fully considered, using the safety targets relevant to each individual *domain*.

Note: the *safety argument* cannot be finalised until it has been shown that the individual *changes* within the *domains* will meet their safety requirements, otherwise the top level safety requirements will have to be re-apportioned to achieve an implementable solution.

6.5.2.2 Claim 2: Logical design satisfies specification and is realistic

This claim focuses on demonstrating safety at the next level of detail. It is here that the assessment looks "inside the box" of the *change* and considers how component parts of the *change* will be designed and interact: it is at this level that the different *domains* of the *TAS* are considered in detail, as well as the interactions and *assurance contracts* between them.

Assessment examines whether the design works as intended under all expected normal and abnormal conditions of the system.

Safety assessment also considers how the elements of the logical design satisfy the overall specification of the change. Failure identification and analysis considers failures of the design elements to deliver their intended function and failures caused by (unintended) interactions between the elements of the design. All such failures are evaluated, by building appropriate models of the system, to determine their effect on the safety of the change and ultimately of the *TAS*. Where failures lead to the *acceptable level of safety* not being achieved, additional requirements need to be introduced to achieve the *acceptable level of safety*. Existing models (e.g. ASCOS SRM) can be useful in analysing these causes and effects, where the model covers the parts of the system being changed. However, generic models become less useful as the assessment extends deeper within

the system because the nature and frequency of failures will be specific to the technology involved. The comments made under Claim 1 (see section 6.5.2.1) on ensuring the validity of any models used are equally applicable here; this is even more relevant where novel solutions are introduced which may not be considered in existing models.

As a result of the assessment, a further level of safety requirements is derived for each element of the design, defining what each design element has to do (both functionality and performance), in order to meet the overall (*TAS* level) safety requirements for the *change*. Assurance requirements are also derived for each design element to define the level of assurance needed that the design elements will meet their requirements.

Interactions between elements of the design are critically important. Suitable techniques should be used to identify and assess these interactions. Where interactions cross boundaries between *domains*, they should be captured in *assurance contracts* agreed between all parties involved.

As described in previous sections, existing techniques and approaches should be considered wherever they are sufficient to deliver the evidence required to support the *claim*. However, it is also important to consider the advantages of harmonising approaches across the *domains*, especially on larger *changes* which span multiple *domains*. (See section 6.4.3.4.)

Note: The implementation is not defined at this stage. However, it does need to be feasible to implement the logical design, and at acceptable cost. Some of the factors which need to be considered are as follows.

- Can equipment / procedures meeting the requirements be produced?
- Can the modifications be implemented / installed to existing equipment?
- Is there a way to transition from current operations to the new state?

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 causes of introduced hazards
- assurance contracts defining the dependencies between domains which need to be satisfied in order to support the argument.

6.5.2.3 Claim 3: Implementation of the logical design is complete and correct

This *claim* focuses on ensuring that the implementation of the designed system meets the requirements. This includes the direct requirements on the individual parts of the system as well as ensuring that the assurance

contracts between different parts of the design are met and that sufficient levels of assurance are generated that the implementation is correct.

The principle aim of safety assessment here is to demonstrate by a combination of analysis and testing, that the (as-built) system[41] implementing the change 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 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 solution architecture.

This claim also derives detailed Safety Assurance Requirements for the solution architecture and shows that these are met. It is at this stage that the change leader often encounters a major problem: test-based techniques are often unable to demonstrate, to a sufficient level of confidence, that the required safety integrity properties of the system have been satisfied. An assurance based approach is often followed to provide this demonstration. (One such approach is defined in the UK CAA SRG CAP670 [37] and the associated AMC [38] for the SW01 requirement.)

Although a large proportion of the work to support claim 3 will be within the individual *modules*, it is also necessary to consider the *assurance contracts* between *modules*. It is likely that some areas will be discovered where the existing *assurance contracts* are not met. In addition, the implementation will make further assumptions about the system and its environment which need to be captured and agreed between the domains. These areas need to be reviewed (see the evaluation process in section 6.5.1.5) and updated accordingly.

For *changes* where equipment is being adapted or developed, the evidence supporting this *claim* will largely be provided by the equipment manufacturer. The *argument architect* will need to review the evidence provided to ensure that it does indeed support the *safety argument* as required.

6.5.2.4 Claim 4: Transition to introduce change is acceptably safe

This *claim* focuses on ensuring that the *change* can be safely introduced into operational service. This is done by showing that

- the fully proven *change* is ready to be brought into operational service
- the process of introduction of the *change* does not adversely affect the overall safety of the *TAS* (e.g. does not cause an unacceptable break in provision of ATM services)

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

[41] Remembering that the system comprises people, processes and equipment.

- **Co-ordination** with all parties affected by the *change*, which may include publication of the details of the *change*
- **Implementation of arrangements for ongoing management**[42] of the changed elements of the *TAS*; where the *change* is in the context of existing service providers, arrangements will already be defined in their management systems, but these may need to be modified to cater for the changes being implemented
- **Assessment of the switchover process** to identify any hazards associated with the switchover process and to introduce any mitigations required to ensure that the safety risk remains acceptable at all times; these mitigations will be part of the arrangements for the switchover
- **Arrangements for the switchover process** for introduction of the *change* - switchover procedures, allocation of responsibilities and the training / briefing of all personnel involved. Where appropriate this should also include fallback / contingency arrangements in case of failures during the switchover process.

For some *changes* (e.g. introduction of a new replacement part) this switchover will be simple and low risk. However for more complex *changes* (e.g. changing the means of surveillance within a particular airspace), especially where multiple stages are involved, the switchover itself is a risky process. These risks should be fully assessed, using a process similar to that used to assess the *change* itself.

This assessment should include full consideration of the human element of the system and their ability to handle the changes. Where changes are wide ranging it may be necessary to stage them so that the operators do not experience a level of change beyond what they are able to handle.

Another issue to consider is where (for example) the *change* is deployed over a period of time such that some parts of the system are operating to pre-*change* requirements / procedures etc, while others are operating to the post-*change* requirements / procedures etc – and to ensure that this does not introduce any unacceptable risks.

Although much of the *safety argument* for this *claim* will be at the level of individual *domains*, it is also critical to ensure that the process is co-ordinated and assessed at the overall *TAS* level. For example, where a new feature / function is being introduced in aircraft operations, it is necessary to ensure that flight crew are properly trained to handle this feature, that crew of other aircraft are properly informed of any effects on their operations, ATM people are properly trained. All must happen before the operations are introduced so everyone knows how to handle the *change*, but not too long before so that those involved have not forgotten their training before the *change* happens.

Primary responsibility for this part of the *safety argument* lies with the operator seeking to introduce the *change*.

[42] including safety management, change management, configuration management

6.5.2.5 Claim 5: Safety monitoring in operational service

Despite all the assessment prior to entry into service, it is impossible to know exactly how the *change* will perform in operation. Assumptions have been made about performance of various elements, about interactions with other parts of the *TAS*; furthermore later *changes* may be introduced which have an (unintended) effect. Thus it is necessary to monitor the *change* to check whether it is safe in service. Where problems are found these need to be assessed and then rectified.

To support this claim, 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 *safety argument*
- processes are in place to report and investigate all safety-related incidents and to ensure that appropriate corrective action is taken in adjusted mitigation/contingency plans
- 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 (extending/limiting a list of potentially affected precursors for a priori risk assessment).

The assessment to support this *claim* should (at least) start during the development of the solution, and not be left to the end of the development. In particular, the identification of metrics to collect in continuous safety monitoring (CSM) may be existing ones already measured within the system or they may be new ones. Identification of the metrics required will be driven by the development of risk models for the change, as developed in support of claims 1 and 2. The ASCOS tool for CSM (see ASCOS D2.5 [7]) can form a useful baseline for the metrics to be collected, supplemented as necessary by further indicators derived from the risk models specific to the change.

When initially submitted for *approval*, the *safety argument* supporting this *claim* necessarily takes a different form from the *safety argument* for the previous claims, because it is about demonstrating that processes are in place, rather than demonstrating that evidence has been collected. In time this is then substantiated with the evidence collected through CSM.

Responsibility for the *safety argument* necessarily transfers to the operator (in collaboration with the *approver*) as they are the ones ultimately responsible for the safety of the system in service. The operator will need support from manufacturer, especially in the analysis of incidents and understanding the impact of those on the safety of the system.

6.6 Obtain Approval

Once the solution and *safety argument* are fully developed, it is necessary to obtain *approval*(s) from the relevant authorities before the change is placed into service. This *approval* will be based on the *approver*'s acceptance of the *safety argument*, the applicable regulations and supporting evidence presented by the

applicant. Note: the *applicant* remains primarily responsible to satisfy themselves that the change is safe (see section 6.4) irrespective of the level of scrutiny from the *approver*.

The role of the *approver*, and the purpose of review by the *approver* is discussed further in section 7.1.3. This section provides guidance to the *approver* on how they can apply risk-based principles to:

- selection of changes to review – section 6.6.1
- the review process itself – section 6.6.2

It is very strongly recommended that the *safety argument* and the proposed evidence to support this argument should have been agreed between *applicant* and *approver* when the *approval plan* is presented (see section 6.4.5). If this agreement is not achieved at the start of the development, there is a significant risk that the argument and evidence produced by the *applicant* are not acceptable to the *approver*. The *applicant* may then need to incur significant extra effort (and significant delay) in order to produce the evidence required. At worst, the *approver* may be completely unable to accept the proposed change.

In addition, a schedule of reviews should be agreed between *applicant* and *approver* when the *approval plan* is presented (see section 6.4.5).

Where a change is split into multiple stages, the *approver* may still insist on reviewing the *safety argument* (and supporting evidence) for all stages before granting any *approvals* in order to avoid the situation where a *change* is partially implemented, but unable to be completed due to lack of adequate argument or evidence for the later stages.

Where a *change* needs *approval* by multiple authorities, *approval* from all relevant *approvers* will be needed before the change is placed into service.

6.6.1 Selection of Changes for Review

The aim of review by the *approver* is to assure that the *acceptable level of safety* is achieved. An *approver* may be selective as to which changes it reviews in detail before granting *approval*. This section considers the factors that should affect the selection of changes for review. (In section 8.3.9, a recommendation is made for further research in this area.

Table 6 outlines the potential safety consequences of the decision whether or not to review a change. These consequences should be borne in mind when developing a selection process for which changes to review.

		Review decision		
		change is reviewed		change is not reviewed
		Review finds issues to be resolved	Review finds **no** issues to be resolved	No findings
True level of safety achieved by the change[43]	safe and adequate *safety argument*	Possibly no direct safety impact; unnecessary changes to *safety argument* or to system; potential changes for the worse	No direct safety impact; if review were really unnecessary, due to too stringent selection, potential long-term damage to culture of *approver*	No safety impact
	safe with inadequate *safety argument*	no safety impact (assuming review correctly finds inadequacies in safety argument, rather than incorrectly questioning safety of system)	indirect damage to *applicant* safety culture	damage to *applicant* safety culture
	unsafe	potentially no safety impact	safety risk in operation	safety risk in operation

Table 6: Safety consequences of review decision

The decision on whether to review a change should be based on:

- the negative safety consequences (A) in the case of the worst possible accident
- the (perceived) probability (B) that the *safety argument* presented will be flawed such that an unsafe change is proposed

Any evaluation of these parameters will always be a rough estimate: it is important to err on the side of caution when making these estimates.

The parameters should be estimated based on:

[43] The term "true level of safety achieved by the change" is used here to distinguish from the level of safety perceived by the *applicant* (as presented in the *safety argument*) and by the *approver*.

- the *approver's* understanding of the likely challenges (C) of the proposed *change*
- the *approver's* knowledge of the *applicant's* organisation, including
  - their technical capability (D) as relevant to the specific *change*[44]
  - their organisational culture (E) as it affects their ability to withstand pressure to make unsafe *changes*

In turn, these are informed by:

- the change definition
- the *approval plan*, including the outline *safety argument* presented therein
- the *approver's* knowledge of aviation safety

Figure 24 illustrates how these factors influence each other.

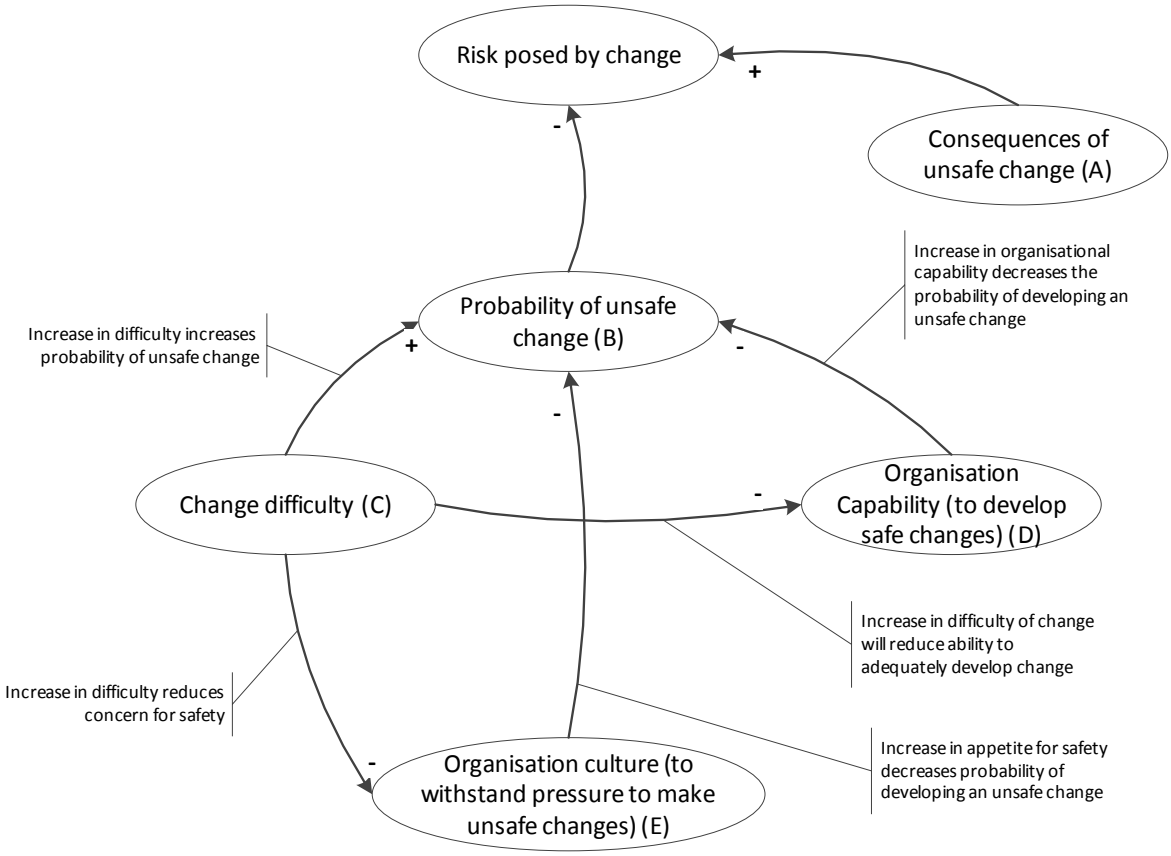

Figure 24: Decision model for risk posed by change

[44] For example, an organisation may have previously demonstrated strong capability to develop a new fixed wing aircraft, but have no experience with rotorcraft.

6.6.2 Reviewing the Safety Argument

This section provides an overview of the process followed by the *approver* to evaluate the *safety argument* being made to support the change.

1. **Familiarisation**: The *approver* gains an understanding of the nature and scope of the change including the stages of implementation. The *approver* also gains an understanding of the structure and organisation of the *safety argument*, including how it will be structured to support the individual stages. As part of this process, the *approver* identifies and records where key topics are addressed to support later assessment activities. As a result, the *approver* forms a view of the scope and adequacy of the *safety argument*. If the *approver* concludes that the *safety argument* is likely to be insufficient, the *applicant* should be informed so that the *approval plan* can be updated accordingly.

2. **Identify the risks**: The *approver* should identify the greatest risks associated with the change in order to prioritise the review effort appropriately. This is determined from the *approver*'s knowledge of:

   - the *applicant*
   - the services it provides
   - the proposed *change*
   - the other organisations involved

   For the lowest grades of risk, the assessment inherently undertaken during Phase 1 may be sufficient to judge the safety of the proposed *change*, so that no further review is required.

3. **Review *safety argument* for the changed system**[45]: The *approver* chooses how to structure and target the assessment to confirm whether the *safety argument* is sufficiently complete and supported by sufficient evidence to show that the risks which are of greatest concern to the *approver* are sufficiently mitigated. If, during the assessment, the *approver* determines that the initial planning was based on an incorrect understanding of the risks associated with the *change*, then the risks are re-assessed (Phase 2) and the assessment plan is revised. The assessment then resumes according to the revised assessment plan.

4. **Determine credibility of planned transitions**[46]: In this stage the *approver* assesses whether the sequence of transitions planned to implement the change is credible, by considering:

   - the feasibility (not safety) of the planned transitional activities that implement the change
   - whether the planned transitional activities are sufficient to implement the stated change
   - whether the prepared parts to be inserted into the *TAS* will be available
   - whether the necessary resources to undertake the change will be available
   - whether there is an adequate safety analysis of the transitional activities
   - whether the criteria to support transition decisions are adequate

[45] This part of the review is focused on the *safety argument* about the final state when the *change* is complete; review of the transitional stages comes later.
[46] The amount of effort required to assess the transitions will depend on their complexity.

This will also allow the *approver* to build an understanding of the transitional activities to support the next phase of the assessment.

5. **Assess safety argument for transition**[46]: In this stage, the *approver* assesses claim 4 of the argument, which relates to transition from the current state of the TAS to the changed state (i.e. when the *change* is completely implemented). For each stage of the transition, the *approver* will need to confirm that:

    - the individual stage is specified to deliver the *acceptable level of safety*
    - a full hazard analysis has been undertaken for the stage to demonstrate that the risk from failures is adequately mitigated
    - the evidence adequately supports the argument
    - appropriate plans are in place and match the full scope of the stage (including installation, commissioning, transition and recovery)
    - the transition activities themselves have been fully assessed and any risks adequately mitigated
    - any uncertainties relating to the implementation of the stage have been identified and addressed as appropriate

    Should any part of the assessment result in significant new information about the risks associated with the change, the appropriate parts of the earlier assessment should be repeated.

6. **Report and address findings**: The *approver* evaluates the concerns recorded during the evaluation to determine their significance in the context of the overall *safety argument*, and the *applicant* is notified of the results. The *approver* must be satisfied with all revisions made to the *safety argument* and supporting evidence to address any identified deficiencies before the change may be implemented.

Where the stages of the *change* are complex and separated in time, the stage-specific assessments may be undertaken separately for the individual stages. This may be driven by the availability of the evidence to support the argument for each stage. *Approval* would then be given independently for the stages as the appropriate assessment is completed.

6.7 Operational Service

Once the change has been granted *approval*, it can be placed into operational service:

- in accordance with the process and timescales agreed with the relevant approvers when they granted approval
- respecting any limitations placed on the operational use, either by the safety argument or by the approval granted

Where a change is staged, each stage must only be placed into operational service once it has gained the appropriate *approval*.

When the change is placed into operational service, the *modules* of the *safety argument*, and especially any *limitations*, effectively become part of the relevant operator's SMS. For example the *module* relating to operation of a new aircraft would become incorporated into the air operator's SMS. However they would still rely on the *assurance contracts* with other *modules* of the *safety argument* continuing to be fulfilled. It is likely that these *assurance contracts* will include dependencies on:

- the maintenance organisation to maintain the aircraft according to the relevant manuals
- the manufacturer to provide updates on component performance
- the crew licensing regime to train flight crew on specific features of the aircraft

The *safety argument* should also be retained to support any further adaptations. Where a change does not directly lead to operational service (e.g. certification of a new aircraft), the context and caveats of the *safety argument* must be included in the relevant certificate. This is necessary so that the *applicant* placing the item into operation can ensure that the item is used in accordance with them.

Monitoring and process improvement must then be undertaken in accordance with Claim 5 of the *safety argument* (see section 6.5.2.5). Note: development of monitoring procedures and KPIs should take place during development as discussed in section 6.5.2.5. Where the *change* is operated by a single operator, this monitoring effectively becomes part of the operator's SMS, and the operator becomes the 'de facto' *argument architect*. It should be noted that some of the inputs for the monitoring may be indicators measured by industry bodies other than the operator. It is essential for accurate safety monitoring that an appropriate level of information is freely available across the *TAS*.

Where the monitoring indicates that the operation of the changed part of the *TAS* may not be as safe as required, then the operator, in conjunction with the relevant authorities, must decide how to address this situation. Initially this may be through more targeted or intrusive monitoring of the system to provide a more detailed assessment. If necessary this is then followed by further changes to the system to ensure the *acceptable level of safety* is achieved – these *changes* would become a new application of the ASCOS Method.

Where different elements of the *change* are operated by different operators, there may be no single owner for the argument. Furthermore, the *TESG* set up to implement the *change*, may be disbanded once the *change* is complete. The arrangements for monitoring such *changes* should be appropriately covered within Claim 5 of the argument, and may require appropriate collaborations to be set up within the *TAS* to ensure that the monitoring, and any corrective action, is carried out adequately.

6.8 Managing Variation and Iteration

Although the steps of the ASCOS Method are shown as a linear progression, it is likely, especially with a complex change, that iteration will be required. It is crucial to ensure that this iteration is properly managed so that the approval path and safety argument remain consistent with the development of the change, allowing the applicant to present a *safety argument* and supporting evidence which are capable of being *approved.*

Development of the argument is iterative for three main reasons:

- **The definition of a *change* may be varied during the lifecycle of the *change*** – this is especially true on large programmes. For example:
    - decisions may be made during development to descope the *change* because part it is infeasible or uneconomic
    - additional requirements may be imposed – for example the need to cater for a new aircraft type
- **Alterations may arise internally** – in the process of developing the *change* it is likely that variations will be needed at multiple levels of the change definition, design and implementation: this may be due to discovering that a particular approach will not work, or is not cost effective, or that the evidence required to support the safety argument cannot be produced.
- **Emerging implementation details give rise to the need for further assurance** – for example, a decision to use a particular type of equipment or process may introduce new hazards which need mitigation through the introduction of further safety requirements.

Any such changes must be evaluated for impact:

- on the later stages of the process
- on the *approval path* and related *safety argument*

The impact of all such variations both on the solution itself and on the *safety argument*, must be properly managed in a controlled fashion so that the solution and *safety argument* remain consistent throughout the lifecycle. This includes examining the impact of variations on development and assessment which has already taken place, and repeating elements of these as necessary. For example, introduction of a new equipment item or process may generate new *introduced hazards* which need mitigation through introduction of further safety requirements.

It is crucial to maintain[47] the *approval path*, *safety argument* and *approval plan* throughout the development lifecycle, modifying them where necessary to remain consistent with the *change* (both with the definition of the *change* and with the solution developed) and with the environment[48]. This also includes modifications to resulting from the evaluation process (see section 6.5.1.5). This may seem obvious, but it is easy for the *approval path*, *safety argument* and *approval plan* to be developed once at the beginning of the lifecycle and then shelved. If they are not maintained during development, the inadequacies[49] which develop will not be noticed until the end of the process, when they are very difficult to rectify.

[47] In this context, "maintain" means "keep up to date with the change to which it applies".
[48] Usually a baseline of applicable regulation is agreed at the outset. However it is possible for new regulations to become applicable within the timescales of the *change*: such developments must be taken into account and their application to the *change* agreed with the relevant *approvers*.
[49] These may arrise due to varations in the solution or due to shortcomings in the evidence produced to support the *safety argument*.

It is also crucial to ensure that any modifications to these items are communicated to all affected stakeholders, especially where assurance contracts are affected by the *change*. Depending on the level of detail of the change, the corresponding parts of the *approval plan* (see section 6.4.5) may also need to be updated and resubmitted to the relevant authorities to ensure that they remain aware of the approach which the *applicant* is taking and the evidence which will be produced. The level of variation which merits representation to the *approver* will be a matter of judgement: the *applicant* should bear in mind that the reason to inform the *approver* is to ease the *approval* process – so if the variation is likely to affect the way in which the *change* is assessed by the *approver*, then it is worth making them aware so that they can plan their *approval* accordingly.

7 Roles and Responsibilities

This section describes the roles and responsibilities involved in applying the ASCOS Method. This is described at an organisational level, however it should be noted that for typical changes different parts of an organisation may take the different roles.

Section 7.1 identifies the roles involved. Section 7.2 shows how these roles are involved at the various stages of the process.

7.1 Roles required within the ASCOS Method

7.1.1 Change Leader

The *change leader* is the organisation with the primary motivation to make the *change* to the *TAS* happen. This organisation will lead the application of the ASCOS Method with support from other organisations as indicated in section 7.2.

The *change leader* is responsible for developing the overall plan for *approval* of the *change*: through the *TESG* (see section 7.1.5) the *change leader* will work with the other stakeholders to ensure that the *change* is developed in a way which is coherent across the whole *TAS*.

The *change leader* is likely to be the organisation introducing the *change* into service and therefore likely to also be (one of) the *applicant*(s). However, they may not be the only *applicant*: for *changes* affecting multiple *domains*, there may be multiple *applicants* (e.g. aircraft manufacturer, air operator, ANSP).

Where a *change* relates to the development and introduction of a new product, especially where a new set of industry-wide requirements is being developed for the product, the role of *change leader* may transfer between organisations during the lifetime of the *change*. For example, the requirements (which may be in the form of a regulation) may be developed by an industry-wide group led by a steering committee drawn from interested organisations (i.e. the *TESG* – see section 7.1.5). The development of specific products (e.g. a specific type of RPAS) may then be led by an individual manufacturer, resulting in the issue of a type certificate. An individual operator will then be change leader for the introduction of individual aircraft into service. This process could, in fact, be viewed as three separate applications of the ASCOS Method, with three separate *change leaders*.

Examples:

- development of a new aircraft, culminating in application for a type certificate would be led by the aircraft manufacturer
- introduction of the aircraft into operation would be led by the aircraft operator, as part of its AOC.
- development of a new surveillance system would be led by the system manufacturer
- introduction of a new surveillance system into operation would be led by the ANSP

- development of a significant new concept (e.g. self-assured separation) would involve multiple *domains* and would need to be led by a group (see *TESG* below) drawing representation from all relevant parts of the industry

For a large part of the lifecycle of the *change*, the *change leader* will also be the *argument architect*. However, once the change enters operational service, the responsibility for the *safety argument* may transfer to another party, hence the use of separate terminology to clarify this.

7.1.2 Applicant

The *applicant* is the organisation which is applying to the *approver* for *approval*.

The *applicant* will be responsible for a *module* of the *safety argument* – the *module* which contains the part of the *safety argument* relevant to the *applicant's domain*. The *applicant* is responsible for ensuring that the *safety argument* within this *module* sufficiently supports the *claim* that, within this *domain*, the *change* achieves the *acceptable level of safety*. The *applicant* is also responsible for ensuring that the *module* satisfies any *assurance contracts* between it and other *modules*. (An *applicant* may be responsible for multiple *modules* if their activities span multiple *domains*.)

For changes involving only one *applicant*, the *applicant* will also be the *change leader*. However, for changes where multiple *approvals* are required, there may be multiple *applicants* within the separate *domains* where *approval* is required.

For example, where the change is for introduction of an RPAS into operation by a specific operator in a particular airspace:

1. certificates of airworthiness for the individual aircraft
2. *approval* for the operator to operate the aircraft
3. *approval* for changes to the ANSP procedures to accommodate the operation of the RPAS
4. *approval* for changes to maintenance procedures to accommodate maintenance of the RPAS

It is likely that the air operator will be the *change leader* and the *applicant* for items 1 and 2. However, there may be other *applicant* for the other items. (Note: it is assumed here that the RPAS has already obtained a type certificate; the manufacturer is likely to be *applicant* and *change leader* for that part of the process.)

7.1.3 Approver

The *approver* is the organisation responsible for approving the *change*. A *change* may involve multiple *approvers*, or multiple disciplines within a single *approver* organisation. Often the *approver* will be an *authority* such as EASA or the relevant national CAA.

The main means by which the *change* is justified to the *approver* is through the *module* of the *safety argument* which relates to the *approver's domain*. The *module* sets out the *claim* that, within the given *domain*, the

change achieves the *acceptable level of safety*. The *approver* also needs to be assured that the *assurance contracts* between this *module* and the rest of the *safety argument* are (and will continue to be) satisfied.

The ASCOS Method does not (directly) affect who is responsible for *approval* of the change; the change needs to be approved by an *approver* in accordance with all the identified applicable regulations and the agreed *approval plan*. (Where the *approval* is a certification, this is also known as the certification basis).

Where multiple *domains* are affected by the *change*, *approval* by all the relevant *approvers* is required before the *change* is put into operation.

As before, with staged changes, the *approver* responsible for *approval* may be different at different stages of the change.

The *approver* will review the *approval plan*(s) received from the *applicant* and change leader to determine whether the proposed approach will lead to a *safety argument* which the *approver* will be able to approve when supported by the appropriate evidence. (Where there are multiple *applicants* and authorities involved, each *approver* will only approve the relevant *module* of the *safety argument*. It is important that, at the planning stage, it is clearly agreed between the *change leader*, *applicants* and *approvers*, that between them they are able to approve all aspects of the proposed *change* and *safety argument*. This must include ensuring that the top level *claim* and the *strategy* for decomposition of this *claim* is acceptable to all *approvers*.)

The *approver* will then approve the relevant *module* of the *safety argument* for the *change* and assure themselves that the *assurance contracts* between the *module* and the rest of the *safety argument* are satisfied, according to the programme agreed in the *approval plan*. The *approver* undertakes this review in order to reduce the probability of an unsafe *change* entering operational service. The *approver* will only approve the *change* if it has been adequately supported by the *safety argument module* presented by the *applicant*. It is not for the regulator to augment the *safety argument* or to provide an alternative *safety argument* in order to *approve* the *change*. *Approval* can only be based upon the contents of the delivered *safety argument*, together with any documented clarifications or further information supplied in response to the *approver*'s enquiries.

7.1.4 Argument Architect

The *argument architect* is responsible for ensuring that the *modules* of the *safety argument*, when taken together, present a complete *safety argument* for the *change* across the whole *TAS*. One of the main tasks here is to ensure that the *assurance contracts* between the *modules* are fully defined and are satisfied by the individual *modules*.

For simple *changes*, the *change leader* may take the role of *argument architect* throughout the lifecycle of the change. Where the *change leader* is also the operator of the changed part of the *TAS*, they may retain responsibility for the *safety argument* once the *change* is introduced to operational service.

For multi-domain *changes*, it is critical that the *argument architect* can view the *change* from the perspective of the overall *TAS* to ensure that the *safety argument* takes into account the requirements of all the *domains*. For such *changes*, it may be necessary to constitute a steering group (the *TESG* – see section 7.1.5) including representatives from the key stakeholder organisations to ensure the *safety argument* is consistent with the needs of all stakeholders.

It should be noted that responsibility for the *safety argument* may change during the lifecycle of the *change*. For example, a specification may be developed by a cross-industry group, and at this stage the *argument architect* may be a *TESG* led by manufacturing organisations. However, the design of a solution to meet this specification would be undertaken by individual manufacturers, each acting as *argument architect* for their own development. (At this stage, the *safety argument* is likely to include proprietary information which stakeholders would not be willing to share across the industry.) A further transfer of responsibility would occur when considering introduction to service, where the *safety argument* would be led by the aspiring operator of the equipment, who will retain responsibility for the *safety argument* once the *change* has been introduced to operational service.

7.1.5 TAS Engineering and Safety Group (TESG)

Where a *change* affects multiple domains, the impact on all *domains* needs to be fully managed throughout the development of the change. It is important to maintain the *safety argument* to be consistent, complete and correct, and aligned to the actual *change*[50]. It is also important to ensure that the interfaces between different parts of the *TAS* (often between *domains*) are managed to ensure that any dependencies are clearly expressed, understood and satisfied. This is especially true where multiple organisations are involved, as it is easy for different parts of the development (and the corresponding *modules* of the *safety argument*) to become out of step.

ASCOS proposes[51] that any complex development should be co-ordinated by a TAS Engineering and Safety Group (*TESG*); the *TESG* would be responsible for co-ordinating all the engineering and safety activities involved in the development of the change. The *TESG* would therefore play the role of *argument architect* for changes involving multiple organisations. The *change leader* (see section 7.1.1) would convene the *TESG* and provide direction: the *TESG* would then ensure that this direction is implemented consistently across the *change*.

Note: as the *TESG* is a co-ordination group and not a legal entity, it would not be able to act as an *applicant* for an *approval*; *approval* would only be granted to a legal entity able to take responsibility for the *change* which is approved. The type of *approval* which as *TESG* might be involved in is one for a jointly developed specification (e.g. stage 1 in Table 3 in section 4.5.1); however the *approval* granted here is of the specification, rather than to an individual *applicant*.

[50] This may seem obvious, but it is easy for the definition of the *change* to be altered during the change lifecycle: it takes good management of the *change* to ensure that the development, assessment and *safety argument* are updated in line with each other and with the definition of the *change*.
[51] This proposal is presented in section 6.3.8 of the WP3 final report [5].

7.1.6 Manufacturer

Where the *change* is (simply) the development and certification of a new product (which may be an entire aircraft), the *change leader* will be the manufacturer of the product (who will also be the *applicant*). However, even in this case, it is likely that the manufacturer will be supplied with parts by other manufacturers.

For other *changes*, the manufacturer(s) will be suppliers to the *change leader*, but will not themselves be either *change leader* or an *applicant*. This is also true in *domains* (e.g. ATM) where products are not subject to *certification*.

The approval-related requirements on the manufacturer should be expressed as *claims* in a *module* of the *safety argument*. The manufacturer will then be responsible for development of this *module* and provision of supporting evidence in order to support these *claims*, and to satisfy any other *assurance contracts* placed on the manufacturer.

In some cases, there will also be providers of services to the *change leader* (e.g. provision of telecomms services): they would be responsible for a *module* of the *safety argument* in the same way.

7.1.7 Affected Organisations

A *change* will usually also affect other organisations not directly involved in the development or *approval* of the *change*.

These are organisations which interface to the changed part of the TAS (e.g. maintainers, pilots, air traffic controllers) and whose activities may be affected by the *change* but where there is no specific *approval* application needed.

These organisations should be included in the consultation process to ensure that any effect on them is fully evaluated and taken into account in the safety assessment.

7.2 Participation within the steps of the ASCOS Method

Table 7 shows the expected involvement of each of the types of organisation described in section 7.1 in the separate steps of the ASCOS Method. Blank cells imply that the organisation has no active involvement.

Step	Organisation					
	Change Leader (supported by *TESG*)	*Applicant*	*Approver*	*Argument architect*	Manufacturer	Affected Organisations
Identify the need	The need for a *change* may be identified by one or more parties across industry: the type of need will then drive which organisation(s) become *change leader*.					
Develop change definition	Lead definition of change at TAS level	Support development of change definition	Support change definition (provide information about requirements and targets)		Provide information about capabilities of products. Support development on concept.	Provide information about impact of change
Develop *approval path*	Lead definition of *approval path*, in collaboration with individual *applicants* where appropriate	Agree *approval plan* with *approver*	Review and accept *approval plan*	Develop *safety argument modules* as required to support *approval path*	Provide information about compliance with requirements	Provide information about impact of change
Develop solution	Lead development of solution at *TAS* level	Detailed development of relevant *safety argument module* and *assurance contracts* and generation of supporting evidence		Monitor compliance with *assurance contracts* between *modules* and ensure that *safety argument* remains complete, consistent and correct across *TAS*	Develop product(s) and services. Supply evidence to support relevant *safety argument modules*	Monitor impact of solution on organisation's domain / operations
Obtain *approval*	Ensure applications for *approval* are co-ordinated and consistent	Make application for *approval*	Review application and grant *approval*		Provide supplementary evidence as required	Provide supplementary evidence as required
Operational Service	Introduce *change* into operation and monitor occurrences of precursor events or other incidents	Responsible for operation under terms of *approval*	Monitor operator's compliance with their SMS	Maintain argument based on monitoring of performance	Investigate occurrences of precursor events or other incidents	Monitor impact of operation on organisation's domain / operations

Table 7: Participation within the steps of the ASCOS Method

8 Conclusions and Recommendations

This sections presents the conclusions of the development of the ASCOS Method and recommendations for further work, as follows:

- Section 8.1 presents a summary of the achievements of this work package to develop the consolidated new approval method, referred to as the ASCOS Method.
- Section 8.2 reviews the ASCOS Method against the principles established earlier in the programme as fundamental to any new method.
- Section 8.3 presents recommendations for further work to improve the ASCOS Method.

8.1 Conclusions

The ASCOS Programme was established to explore the need for adaptation of existing *approval* processes in response to:

- fundamental changes in the institutional arrangements for aviation regulation in Europe
- the introduction of new technologies and operations
- demands for higher levels of safety performance

The objective of the programme was to develop novel *approval* processes and supporting design methods and tools to ease the *approval* of safety enhancement systems and operations. The programme was tasked with providing a method which delivers:

- 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

Initial activities reviewed current *approval* processes and chose four options for improvement as well as a set of principles to be adopted by the new ASCOS Method. The initial proposal for an ASCOS Method (published as ASCOS D1.3 [3]) comprised eleven steps based around the construction of a *safety argument* to support the *claim* that the change made to the *Total Aviation System (TAS)* would achieve the defined *acceptable level of safety*.

Following the publication of D1.3 [3], the eleven step method was applied to four case studies representing possible *changes* to the *TAS*. The aim was to exercise the method and provide feedback to improve it. Although these case studies struggled in their application of the steps, they yielded very useful feedback, both in written form and through the involvement of the authors of the D1.3 method. A comprehensive set of recommendations has been published [39] based on the results of the case studies and on separate validation exercises undertaken with the ASCOS User Group. These recommendations have been used to refine the ASCOS Method into the form presented in this report.

The consolidated ASCOS Method presented in this report focuses on establishing an *approval path* for a change to the *TAS* and then providing support for following that *approval path* through the lifecycle of development and deployment of the *change*. The ASCOS Method is presented as a framework of activities which can be adapted and iterated as required, rather than a rigid process of sequential steps.

The ASCOS Method uses existing approaches which are adapted and augmented only when necessary. (This may be to accommodate innovation, to ensure interfaces are managed or simply to streamline the process.) The ASCOS Method provides a framework for development of such adaptations, which provides support throughout the lifecycle, starting with identification of the concept and establishing its viability, through development and implementation into operation and sustainment. However, the activities do not depend on a particular lifecycle being followed. The ASCOS Method is not just applicable to *certification*; it is also applicable to more general *approvals*.

The *approval path* is supported by development of a modular *safety argument* to support the claim that the *acceptable level of safety* is achieved by the change to the *TAS*. The *safety argument* is presented as a hierarchical set of *claims*, supported by evidence and is developed to consider all aspects of the *TAS* affected by the *change*. The *safety argument* is partitioned into *modules*, each containing the *safety argument* relevant to an individual *domain* of the *TAS*. Dependencies between these *modules* are expressed as *assurance contracts* agreed between the owners of the *modules*.

The modular structure of the *safety argument* allows the *modules* to be developed separately by the stakeholders in the individual *domains* in confidence that the final result will be a consistent, complete and correct overall *safety argument*. This structure also allows clear separation between the parts of the *safety argument* which need to be *approved* by the different *approvers* involved. However, this also introduces a significant risk of divergence between the *modules* in ways which were not envisaged when the *modules* were created. It is therefore necessary to ensure that the argument is properly maintained and integrated throughout the development by an *argument architect*.

The structure of the *safety argument* can be presented in a graphical form (e.g. Goal Structuring Notation (GSN)) to aid understanding, although it is always supported by text which explains what is being claimed. The logical argument uses the same basic concepts as the SESAR Safety Reference Material (SRM) [15] (in turn based on EUROCONTROL Safety Assessment Method (SAM) [9] and Safety Assessment Made Easier (SAM-E) [27] approaches): these have been developed to provide specific guidance for application across the *TAS*. The ASCOS Method provides flexibility to encompass novelty and innovation, while also allowing existing methods and approaches to be retained where appropriate. It also supports the evaluation of the context within which these existing approaches operate, in order to establish whether they need adaptation and, importantly, to record the rationale for such decisions.

The ASCOS Method recognises the significant underlying differences in approach between *domains*, including levels of safety, assessment methods and terminology, sometimes giving significantly different meanings to the same term. Differences between *domains* are understandable given the structure and history of the

different parts of the *TAS*, but careful consideration is therefore needed in building an integrated method. The ASCOS Method does not in itself mandate how safety targets for a *change* should be established, but recognises that the current high level of safety must be maintained. It is usually not practical to trade off safety between *domains* because it is difficult to justify a decrease in safety in any one *domain*. To do this, it would be necessary to provide a robust quantification across all domains which demonstrates a significant overall positive impact on safety. Production of such a robust quantification is made more difficult by the fact that different *domains* use different types of targets (often with different units), making it difficult to create valid comparisons between *domains*. A corresponding assessment would be needed in the event of a *change* with differing impacts on different sovereign states.

The ASCOS Method also addresses the difference between performance based and compliance based approaches. The ASCOS Method allows goal based safety arguments (a performance based approach) using high level, solution independent targets to support the development and assessment of innovative solutions, while also allowing more detailed requirements to be used to ensure consistent application of established solutions. Prescriptive requirements (a compliance based approach) are also useful to constrain interfaces or express well established rules, especially where these relate to interfaces with parts of the *TAS* unaffected by a *change*.

Co-ordination between all parties involved in the *change* is critical to successful and efficient implementation. This is reflected in:

- early engagement between all stakeholders (including the *approver*), resulting in production and agreement of an *approval plan*, based on the *safety argument*, which guides the generation of the evidence needed to support the *approval*
- the use of *assurance contracts* to record and manage dependencies between stakeholders, allowing the *safety argument* to be divided into *modules* to be supported by individual stakeholders, giving freedom in their substantiation of the *safety argument*, as long as the *assurance contracts* are satisfied
- the establishment, where appropriate, of a steering committee (the *TESG*) for development and assurance of the *change*, with representatives drawn from all the relevant organisations and disciplines

Guidance is provided in this report to show how the *safety argument*, and the activities, should be adapted according to the needs of an individual *change*. This recognises that although the overall concept can be applied to any *change*, the actual *safety argument* required will vary widely depending on the particular change to be made – for example, the *safety argument* for introduction of a new equipment item on an aircraft will be very different from the *safety argument* for a change to the arrivals concept at a particular aerodrome.

Application in the Case Studies, supported by the validation exercises, shows that the ASCOS Method is capable of analysing and demonstrating acceptable safety for new concepts and technologies, considering the

entire *TAS* rather than sub-elements in isolation, therefore delivering two of the three objectives set in the ASCOS remit. However, it is difficult to introduce the flexibility to accommodate innovation and to address changes which span the *TAS* (the second and third objectives above) without having a negative impact on the cost and efficiency of the approval process, at least in the short term. In addition, the innovations envisaged within aviation may also drive up the scale and complexity of the safety assurance required, having a further negative impact on the efficiency of the approval process, especially given the limited availability of expert safety assurance resource. However, this barrier needs to be overcome in order to realise the significant operational, financial and safety benefits which are available and which outweigh the increased cost of safety assurance. In addition, there was consensus within the ASCOS analysis that cost and efficiency of the assurance will improve in the medium and longer term as the ASCOS Method becomes established within the community.

Inevitably, further improvements and refinements are possible. A list of recommendations is presented in the next section of this report.

However, the greatest opportunity for improvement will come from application of the ASCOS Method in earnest in real situations. The ASCOS Consortium therefore commends this ASCOS Method to EASA for adoption as a means of establishing *approval* for changes to the Total Aviation System within Europe.

8.2 Assessment

As discussed in section 1.2, earlier work within this ASCOS work package identified a series of principles to be employed by any new approval method. Table 8 reviews how these principles have been addressed in the ASCOS Method.

Principle	Means of Addressing
Avoid unnecessary change, recognising the good approaches already in place	*Approval path* adopts existing approaches where appropriate with adaptation and / or augmentation where necessary to support innovation
Provide a generic certification framework encompassing the Total Aviation System (TAS)	*Safety argument* composed of *modules* encompasses whole *TAS* and establishes *assurance contracts* between the separate *domains* and organisations as necessary.
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	Standard terminology has been adopted and is explained in Appendix A.

Principle	Means of Addressing
Provide rigorous management of interfaces, both between domains and between the TAS and its environment, to ensure that all key safety issues are properly addressed and not lost at interfaces	*Assurance contracts* between *modules* established as a means of documenting, agreeing and monitoring issues across interfaces.
Allow, within each domain, the new method to evolve from current approaches by keeping the existing approach where no change is required, learning lessons from other domains where this gives improvement and ensuring that bottlenecks and shortcomings are addressed by the proposed approach.	Addressed by basing the approval for a given change on the existing approval path within the domain, adapted or augmented as necessary to refine the approach.
Promote flexibility within each domain to allow introduction of new technologies or procedures	Flexibility provided by the use of a safety argument framework which allows for new approaches to be developed where necessary to encompass innovation.
Harmonise approaches between domains where this is advantageous or necessary	Framework provided by the ASCOS Method described in this document; harmonisation of assessment processes and standards addressed in a separate ASCOS work package and reported in D3.6 [5]
Simplify existing processes, where there are demonstrable benefits and no loss of confidence in the assurance of safety	Simplification and harmonisation of detailed processes was addressed in ASCOS WP3 (see D3.6 [5]).
Reinforce existing techniques where they are appropriate but not consistently applied	Development of the ASCOS Method has not explicitly considered reinforcement of existing techniques, but this could be a side effect of the evaluation of techniques which forms part of the ASCOS Method (see section 8.3.1).
Provide a mechanism for identification and resolution of further bottlenecks and shortcomings	Development of the *approval path* required by the ASCOS Method includes steps to review the existing approaches used in the domain to identify inefficiencies and to refine / revise them as necessary
Introduce a bridge between the regulations in different domains where needed, in particular between aircraft certification and Air Traffic Management	*Assurance contracts* between *modules* established as a means of documenting, agreeing and monitoring issues across interfaces.

Principle	Means of Addressing
Take into account the electronic hardware more explicitly in the proposed approach	The purpose of the ASCOS Method is to develop a framework across the *TAS*; at the level of such a framework it is not appropriate to address specific concerns relating to the assurance of electronic hardware. (A recommendation for further research in this area is made in section 8.3.8.)
Consider the fact that less experience is gained by the flight crew when more automation is used	ASCOS Method includes framework for assessing the impact of a change across the TAS, including any unintended or unforeseen changes.

Table 8: Assessment against principles established for development of new approval method

8.3 Recommendations

This section contains recommendations for work which would improve or support the ASCOS Method. (Note: we have included recommendations from other ASCOS reports only where they are pertinent to the conclusions of this report.)

8.3.1 Adoption of ASCOS Method

The ASCOS Consortium recommends adoption of the ASCOS Method as the method to be used when making *changes* to the *TAS*.

The ASCOS Method can be applied to any *change*; where *changes* are sufficiently routine and their effects are contained within a single *domain*, an early evaluation will establish that the *approval path* for the change relies solely on existing approaches and requires no further adaptation. The lessons learned from application of the ASCOS Method should be used to further refine the method.

The ASCOS Method includes steps to evaluate existing techniques to establish whether they remain appropriate for development of innovative solutions; this evaluation could also be used to reinforce existing techniques where they remain appropriate.

8.3.2 Documentation of Implicit *Safety Arguments*

The ASCOS Consortium recommends documentation of the implicit *safety arguments* currently followed in the individual domains. *Safety arguments* are often implicitly defined by the *approval* process followed in individual domains. Documentation of implicit *safety arguments* will make it easier to develop robust *safety arguments* for changes where the existing *approval* path needs to be modified to accommodate the change.

8.3.3 Sharing of Safety Risk Information

The ASCOS Consortium recommends that the EC or EASA promotes the sharing of safety risk information between *TAS* stakeholders. The success of the ASCOS Method depends critically on establishing open

communication between stakeholders involved in a particular *change*. However, the exchange of proprietary information is often blocked by an organisation's legal department because of concern that information may either damage an organisation's reputation or may put them at a competitive disadvantage. It is therefore unlikely that such information will be freely shared without some promotion and / or enforcement by the EC or by EASA.

(This recommendation was previously proposed in the ASCOS Validation Results [39] (REC1.09) where further details can be found.)

8.3.4 Definition of *Domains*

The ASCOS Consortium recommends that the definitions of the individual *domains* of the *TAS* should be further refined, taking into account both the EASA regulatory structures and the operational structures within the *TAS*.

8.3.5 Refinement of TESG Concept

The ASCOS WP3 final report [5] proposes the establishment of a *TAS Engineering and Safety Group (TESG)* for any complex *change*. This *TESG* would be responsible for co-ordinating the engineering and safety activities involved in the development of a *change*. This is a very important role in ensuring that the interfaces between stakeholders are properly established and that open communications are possible throughout the lifecycle of a change. The concept of a *TESG* is also strongly related to the concept of an *argument architect*, which is critical to the ASCOS Method.

The ASCOS Consortium recommends that further research is undertaken into how *TESGs* could be established and how they could fulfil the role of *argument architect* for complex changes. This research should further develop the remit already proposed by ASCOS WP3 [5].

Success of the *TESG* concept is also dependent on the establishment of open communications as covered in the separate recommendation in section 8.3.3.

8.3.6 Example *Safety Arguments*

The ASCOS Method intentionally provides an adaptable framework for developing *approval* paths and *safety arguments* for individual changes. Effort has focussed on establishing the framework which forms the ASCOS Method; it has not been possible to develop detailed applications of this framework to multiple real applications. In addition, it is fundamental to the ASCOS Method that existing safety assessment approaches are utilised as far as possible, and adapted or augmented only where necessary. The case studies provided valuable feedback to refine the method, but did not yield detailed examples of end-to-end application of the final method. In addition ASCOS WP3 [5] proposed a unified framework for safety assessment processes across all *domains* of the *TAS*.

The ASCOS Method would significantly benefit from being complemented by detailed worked examples for a variety of types of *change* and *domains*, showing how the existing approaches are evaluated, incorporated and adapted or augmented. These examples could be extended as the ASCOS Method is applied to an increasing number of changes and form an ever expanding of repository of guidance for application of the method.

However, it is noted that these examples should never be viewed as templates which can be simply picked up and applied without intellectual effort: the temptation to do this will be great, but each change will be different and will need detailed consideration on its own merits.

8.3.7 Trade-Off of Safety Between *Domains*

The research undertaken by ASCOS has highlighted the difficulty of justifying changes where, although there is a significant safety benefit overall, there is a safety disbenefit in one *domain* of the *TAS*. As discussed in sections 2.5 and 6.3.8, such *changes* require robust justification demonstrating a significant overall positive impact on safety. Such justifications are made more difficult by the fact that individual *domains* use different units and means of measurement.

The ASCOS Safety Risk Model (see D3.6 [5]) has built on previous work and has made steps towards establishing a way in which such comparisons between *domains* can be made. However, this model is not yet a sufficiently mature and complete model of the whole *TAS* to form a basis for the robust justifications needed to trade off safety between *domains*.

The ASCOS Consortium recommends further research in this area in order to move towards a situation where it is possible to trade off safety between *domains* and thus support the approval of *changes* which deliver an overall benefit to safety where there is a (small) disbenefit in a single *domain*.

8.3.8 *Certification* of Electronic Equipment

The principles identified early in WP1 included a desire to take electronic hardware more fully into account. As explained in section 8.2, it was not possible to consider this detailed concern within the development of the ASCOS Method.

Certification of such equipment is well-established in the aircraft *domain* and in parts of other industries (e.g. railway signalling). It is perceived by other *TAS domains* (e.g. ATM) that introduction of *certification* would make it easier to develop and deploy such equipment.

The ASCOS Consortium recommends further research into the introduction of *certification* for electronic equipment across the *TAS*, with a particular emphasis on equipment used for ATM. This research should pay particular attention to ensuring that the *certification* approach recognises the importance of the environment within which equipment is used and the need to evaluate this for each application. (For example, it is necessary to examine the effect of use with different operating procedures, different types of air traffic and different traffic volumes.) The research should also consider that, where the market in equipment is relatively small (e.g. in ATM), the cost of any *certification* scheme to the manufacturers must be kept low enough to

ensure that suppliers are able to implement it without leading to unacceptable increases in the price of the equipment.

8.3.9 Selection of *Changes* for Review

Review of a *change* by the *approver* is a key part of the *approval* process; it is important that the *approver* selects appropriately which *changes* should be reviewed. Some guidance is presented in section 6.6.1 based on criteria used by the UK CAA. However, further research in this area may benefit *approvers* by enabling them to concentrate resources on the *changes* most needing their attention.

The ASCOS Consortium recommends further research into the factors which affect the development of safe *changes* in order to support *approvers* in making decisions about how these *changes* should be reviewed.

References

#	Authors(s), Title, Year
1	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.
2	EC, Flightpath 2050: Europe's Vision for Aviation, Report of the High Level Group on Aviation Research, ISBN 978-92-79-19724-6, 2011.
3	ASCOS, Outline Proposed Certification Approach (D1.3 Version 1.2), 2013
4	ASCOS, WP1 Final Report (D1.6), 2015
5	ASCOS, WP3 Final Report Safety Risk Management (D3.6 Version 1.2), 2014
6	EUROCONTROL, European Operational Concept Validation Methodology (E-OCVM Version 3.0) Volume 1, 2010
7	ASCOS, WP2 Final Report Continuous Safety Monitoring (D2.5 Version 1.3), 2014
8	EASA, CS-25 Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes, Amendment 17 (Annex to ED Decision 2015/019/R), 2015
9	EUROCONTROL, Safety Assessment Methodology (SAF.ET1.ST03.1000-MAN-01 Edition 2.1), 2006
10	EASA, Composite Aircraft Structure (AMC20-29 - Annex II to ED Decision 2010/003/R), 2010
11	Journal of Aviation Management: Safety Management, Certification and the Extended Use of Composite Materials in Large Passenger Aircraft Structures, 2014
12	Origin Consulting, GSN Community Standard, Version 1, 2011
13	EUROCONTROL, Safety Case Development Manual (DAP/SSH/091), 2006
14	UK CAA Guidance on the Conduct of Hazard Identification, Risk Assessment and the Production of Safety Cases (CAP 760 First Edition, Amendment 2010/01), 2010
15	SESAR: Safety Reference Material, Edition 00.02.01, Project ID 16.06.01, 30th Jan 2012
16	Dr TP Kelly COMSA/2001/1/1: Concepts and Principles of Compositional Safety Case Construction, 2001
17	Charles Haddon-Cave QC, The Nimrod Review – An independent review into the broader issues surrounding the loss of the RAF Nimrod MR2 Aircraft XV230 in Afghanistan in 2006, 2009
18	OPENCOSS, Baseline for the Common Certification Language (D4.1 Version 1.0), 2012
19	William S. Greenwell et al, A Taxonomy of Fallacies in System Safety Arguments, 2006
20	D.N. Walton, Reasoned Use of Expertise in Argumentation, Argumentation Vol. 3, pp.59-73, 1989.
21	Hahn and Oaksford, A Bayesian approach to informal argument fallacies, Synthese (2006) 152 pp 207 – 236
22	EASA, European Aviation Safety Plan 2014-2017, 2014
23	EASA, Opinion 03-2014: Requirements for service providers and the oversight thereof, 2014

#	Authors(s), Title, Year
24	RTCA, Guidelines for Approval of the Provision and Use of Air Traffic Services Supported by Data Communications (DO-264 also known as ED-78A), 2000
25	FAST, Areas of Change, http://www.nlr-atsi.nl/fast/aoc, Accessed 26th August 2015
26	UK Parliament, Health and Safety at Work Act, 1974
27	EUROCONTROL Safety Assessment Made Easier – Part 1 Safety Principles and an Introduction to Safety Assessment (Edition 1.0), 2010
28	EASA, Certification Specifications for Small Rotorcraft (CS-27 Amendment 3 - Annex to ED Decision 2012/021/R), 2012
29	JARUS, Certification Specification for Light Unmanned Rotorcraft Systems (CS-LURS), Version 1.0, 2013
30	EUROCAE, Guidelines for Development of Civil Aircraft and Systems (ED-79A), 2011
31	EUROCAE, Software Integrity Assurance Considerations for Communication, Navigation, Surveillance and Air Traffic Management (CNS/ATM) Systems (ED-109A), 2011
32	EUROCAE, Process for specifying risk classification scheme and deriving safety objectives in ATM (ED-125), 2010
33	RTCA, Software Considerations in Airborne Systems and Equipment Certification (DO-178C), 2011
34	IEC, Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems (IEC 61508)
35	RTCA, Environmental Conditions and Test Procedures for Airborne Equipment (DO-160), 2010
36	ASCOS, Evaluation of certification case studies (D4.5 Version 1.0), 2015
37	UK CAA, Air Traffic Services Safety Requirements (CAP 670 Version 3 / Amendment 1), 2014
38	UK CAA, Acceptable Means of Compliance to CAP 670 SW 01: Guidance for Producing SW 01 Safety Arguments for COTS Equipment (Issue 3), 2010
39	ASCOS, Validation Results (D5.4), 2015
40	ICAO, Safety Management Manual (ICAO 9859, Third Edition), 2013
41	ISO / IEC, Systems and software engineering —Systems and software assurance (BS ISO / IEC 15026)
42	OMG, Structured Assurance Case Metamodel (SACM), Version 1.0, 2013
43	Ebeni, Summary Report – Safety Assurance of the Draft Specifications for the Use of Military UAVs as OAT Outside Segregated Airspace, 2006
44	EUROCONTROL, The EUR RVSM Pre-Implementation Safety Case (RVSM 691 Version 2.0), 2001
45	Fowler D., Getting to the Point: A Safety Assessment of Arrival Operations in Terminal Airspace, Air Traffic Control Quarterly Volume 20 Number 2, 2012
46	Thomas S., Fowler D., (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

#	Authors(s), Title, Year
47	Wagner S. et al, A Case Study on Safety Cases in the Automotive Domain: Modules, Patterns and Models, 2010
48	Bates S. et al, Safety case architectures to complement a contract-based approach to designing safe systems, 2003
49	Fenn J. et al, Safety Case Composition Using Contracts – Refinements based on Feedback from an Industrial Case Study, 2007
50	CENELEC, Railway applications – Communication, signalling and processing systems – Safety related electronic systems for signalling (EN50129:2003)
51	ISO, Road vehicles – Functional Safety – Part 2: Management of Functional Safety (ISO 26262-2:2011)
52	EC, 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 (EC748/2012), 2012
53	EASA, ETSO Authorisations (http://www.easa.europa.eu/certification/ETSO-authorisations.php), accessed 13/12/2013.

Appendix A Terminology Reference and Abbreviations

Terms which have specific meanings within the ASCOS Method are defined in Table 9. Where these terms are used in this document they are shown in *italic* type.

Where possible terms have been given the same meaning as they have across the European aviation industry and a different term is used where a different meaning is intended. Where this has not been possible, this is highlighted within the definition given in the table.

Abbreviations used in this document are listed in Table 10.

Term	*Meaning*	*Related term(s)*
acceptable level of safety	the level of safety which the *approver* requires the *change* to achieve. Note that it may be acceptable for the *change* to maintain the existing level of safety. (See sections 2.5, 6.3.8.)	
applicant	responsible for making the application to the relevant *approver* for *approval* of a *change* (or part thereof) to the *total aviation system (TAS)*. For an *operational change*, this will usually be the organisation which is putting the *change* into operational service; for other changes (e.g. *certification* of a new product or aircraft) this will usually be the manufacturer. (See section 7.1.2.)	
approval	declaration by the *approver* that the *change* meets the set of requirements, including the *acceptable level of safety*, agreed in the *approval plan*. For an *operational change*, this is the permission required before the change can be placed into *operational service*. The term *approval* is used in this document to differentiate from *certification* which often has a narrow interpretation of certifying a product to a specific set of (generic) requirements. (See section 2.1.) (*Approval* is the term used in the recently proposed EASA IR on oversight of (air traffic) service providers [23].)	approval path, approval plan, approver, certification
approval path	the approach followed by the *applicant* to gain *approval* for the *change*; this will follow existing established approaches where possible, but these may be adapted or augmented by new approaches where necessary to accommodate innovation or to ensure that the approach addresses the whole *TAS*. (See section 3.2.)	approval, approval plan, approver

Term	*Meaning*	*Related term(s)*
approval plan	document in which the *applicant* sets out the *safety argument* for the *change* and describes how and when the supporting evidence will be produced. The *approval plan* is agreed between *applicant* and *approver* early in the lifecycle and forms the basis for the *approver*'s review of the *applicant*'s submissions.	approval, approval path, approver
approver	responsible for approving a *change*; for an *operational change* this gives permission for it to be placed into operational service. For many changes the *approver* will be the relevant *authority*; however not all changes require *approval* by an *authority*. (See section 7.1.3.)	approval, approval path, approval plan
argument architect	responsible for constructing and maintaining the *safety argument* for the *change*. In particular the *argument architect* will focus on ensuring that, where multiple organisations are involved, their contributions, when taken together form a complete, consistent and correct *safety argument*. (See section 7.1.4.)	safety argument
assumption	a statement which is believed to be true and which is assumed to be true for the purpose of the *safety argument* but which is not (yet) supported by evidence	caveat
assurance contract	definition of interface between *modules* of the *safety argument*; intention is that the owner of the *module* has freedom in developing it, as long as the *assurance contract* is satisfied	module
authority	an agency or body created by a government and provided with institutionalized and legal power to perform a specific function; in this context of the ASCOS Method, this is used to refer to an organisation competent to approve changes to a particular part of the *TAS*; the *approver* of a *change* will often, but not always, be the *competent authority* in that system domain	approver, competent authority
caveat	something which must be taken into account when considering the conclusions of the safety argument. This is a general term encompassing *assumptions, conditions, constraints, limitations* and *safety issues*.	assumptions, conditions, constraints, limitations and safety issues
certification	any form of recognition by a competent authority that a product, part or appliance, organisation or person complies with the applicable requirements (See section 2.1.)	approval
certification basis	agreed set of standards with which an *applicant* has to demonstrate that the subject item (or organisation) is compliant in order for the *approver* to grant *certification*	certification

Term	*Meaning*	*Related term(s)*
change	any alteration to the *TAS*, beyond intended operational use or maintenance. The purpose of applying the ASCOS Method is to obtain *approval* for a *change* to the *TAS*.	
change leader	the organisation primarily motivated to introduce the *change*.	
claim	a proposition (true or false statement) which is asserted as part of the *safety argument*	safety argument
competent authority	an *authority* with the inherent competence in a specific area; the *approver* of a *change* will often, but not always, be the *competent authority* in that system domain	authority, approver
condition	something which must be fulfilled before a *claim* is valid	caveat, assumption, safety issue, limitation
constraint	a restriction in the design or integration of components required for a *claim* to be valid	caveat, assumption, condition, safety issue, limitation
domain	one of the subparts of the *TAS*; the term *domain* is not precisely defined in this document and a proposed area of further work is to provide a rigorous definition of the *domains* of the *TAS* (See Appendix B.)	total aviation system
hazard	a condition which could cause or contribute to unsafe operation within the *TAS* (Adapted from ICAO SMM [40] section 2.13.2.)	introduced hazard, inherent hazard
inherent hazard[52]	a *hazard* which is present in the *TAS* before the introduction of the change	hazard, introduced hazard
introduced hazard[52]	a *hazard* introduced to the *TAS* as a result of the change, for example due to a failure of a component introduced by the change	hazard, inherent hazard
limitation	a restriction on the (scope of) deployment and / or operation of the *change*. (From EUROCONTROL Safety Case Development Manual [13])	caveat, assumption, safety issue, condition
logical design	a definition of the *change* at the level of machine-based functions, human roles and tasks, but not defining the specific equipment, procedures or training	
module	a subdivision of the *safety argument*, related to other modules by means of *assurance contracts*	safety argument, assurance contract

[52] The distinction between *inherent hazards* and *introduced hazards* is made mainly in order to highlight that (a) there are *hazards* already in the *TAS* before any *changes* are introduced and (b) *changes* can, in themselves introduce *hazards*. It is more important to ensure that all *hazards* are identified and mitigated, than to worry about classifying them correctly.

Term	*Meaning*	*Related term(s)*
operational change	a *change* which, when introduced into operational service, will directly affect the *TAS*; this should be contrasted with certification of a new product (including an aircraft), which does not directly affect the *TAS*, until it is introduced into operational service.	change
safety argument	a logical argument formed from a connected set of *claims*, supporting information and evidence used to persuade the reader that the proposed *change* will achieve the defined *acceptable level of safety*.	
safety issue	an issue which must be resolved before a *claim* can be considered to be valid	caveat, assumption, limitation, condition
strategy	an element of the *safety argument*, explaining how a parent *claim* is achieved by the supporting sub*claims*	claim
total aviation system (TAS)	the whole aviation system (See Appendix B.)	domain

Table 9: Terms used with specific meanings in D1.5

Abbreviations	*Description*
AARS	Automated Aircraft Recovery System
ACARE	Advisory Council for Aviation Research and Innovation in Europe
ACAS	Airborne Collision Avoidance System
AIP	Aeronautical Information Publication
AIS	Aeronautical Information Service
AltMoC	Alternative Means of Compliance
AMC	Acceptable Means of Compliance
ANS	Air Navigation Service
ANSP	Air Navigation Service Provider
AOC	Air Operator Certificate
AoC	Area of Change
ASCOS	Aviation Safety and Certification of new Operations and Systems
ATCO	Air Traffic COntroller
ATM	Air Traffic Management
ATS	Air Traffic Services
CAA	Civil Aviation Authority
CATS	Causal model for Air Transport Safety
CENELEC	European Committee for Electrotechnical Standardization

Abbreviations	*Description*
CFIT	Controlled Flight Into Terrain
CofA	Certificate of Airworthiness
CS	Certification Specification
CSM	Continuous Safety Monitoring; Common Safety Method
EASp	European Aviation Safety Plan
EASA	European Aviation Safety Agency
EC	European Commission
EFB	Electronic Flight Bag
E-OCVM	European Operational Concept Validation Methodology
ESD	Event Sequence Diagram
ETSO	European Technical Standard Order
EUROCAE	European Organisation for Civil Aviation
EUROCONTROL	European Organisation for the Safety of Air Navigation
FAST	Future Aviation Safety Team
FMS	Flight Management System
FTA	Fault Tree Analysis
GSN	Goal Structuring Notation
ICAO	International Civil Aviation Organization
IMA	Integrated Modular Avionics
IR	Implementing Rule
ISO	International Organization for Standardization
JARUS	Joint Authorities For Rulemaking of Unmanned Systems
LOC-I	Loss Of Control - Inflight
LURS	Light Unmanned Rotorcraft Systems
MCC	Means of Compliance Checklist
OMG	Object Management Group
OPENCOSS	Open Platform for Evolutionary Certification of Safety-Critical Systems
RF	Radio Frequency
RNP	Required Navigation Performance
RPAS	Remotely Piloted Aircraft System
RTCA	Radio Technical Commission for Aeronautics
RVSM	Reduced Vertical Separation Minima
SACM	Structured Assurance Case Metamodel
SAM-E	Safety Assessment Made Easier
SARPS	Standards and Recommended Practices
SCDM	Safety Case Development Manual
SEooC	Safety Element out of Context

Abbreviations	*Description*
SESAR	Single European Sky ATM Research
SMM	Safety Management Manual
SMS	Safety Management System
SPI	Safety Performance Indicator
SRAC	Safety Related Application Condition
SRG	Safety Regulation Group
SRM	Safety Reference Material; Safety Risk Model
TAS	Total Aviation System
TC	Type Certificate
TESG	TAS Engineering and Safety Group
UAV	Unmanned Aerial Vehicle
WP	Work Package

Table 10: Abbreviations used in this document

Appendix B The *Total Aviation System (TAS)*

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 system where all the parts interact. It is thus important to consider the impact of a change on the whole system, and to do this it is important to understand the parts of the system and how they interact. It is particularly important to understand the interfaces between the parts of the system, as it is here at the interfaces where interactions can easily be overlooked or misunderstood, leading to potential safety problems.

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

The Total Aviation System can be defined at a number of levels, including:

a. functional specification, including high level functions, performance, operational behaviour and modes of operation;

b. logical design: a high-level architectural representation of the system, independent from the 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. implementation: the details of equipment (hardware, software and data), people (flight crew, controllers and maintainers), operation and maintenance procedures, training and sectorisation.

One of the concepts introduced in the ASCOS Method is to subdivide the TAS into domains, and there are different ways in which this can be done. As the ASCOS Method is about *approval* of changes, there is merit in aligning these domains to the structure of the applicable regulations, as portrayed in the EASA Regulations Structure, as shown in Figure 25. (Some of the illustrations in the body of this document use an adapted version of this structure.) However, there are aspects of the TAS (e.g. manufacturers, especially of non-airborne equipment) which are not clearly visible in this structure: it is sometimes useful to use a functional breakdown, as shown in Figure 26.

One recommendation for further research (see section 8.3) is to develop a subdivision of the TAS which is aligned to the EASA regulation structure but which also captures the relevant interactions between the parts of the system.

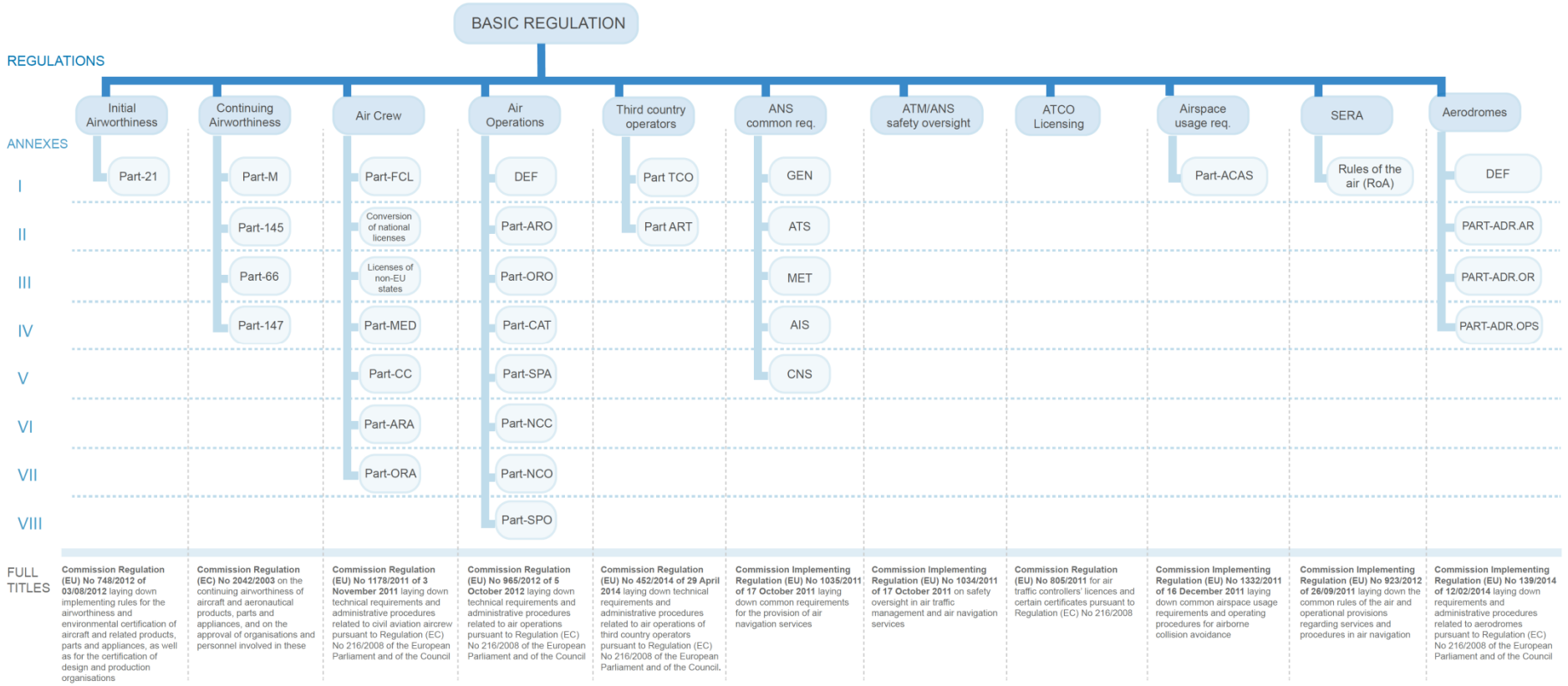

Figure 25: EASA Regulations Structure

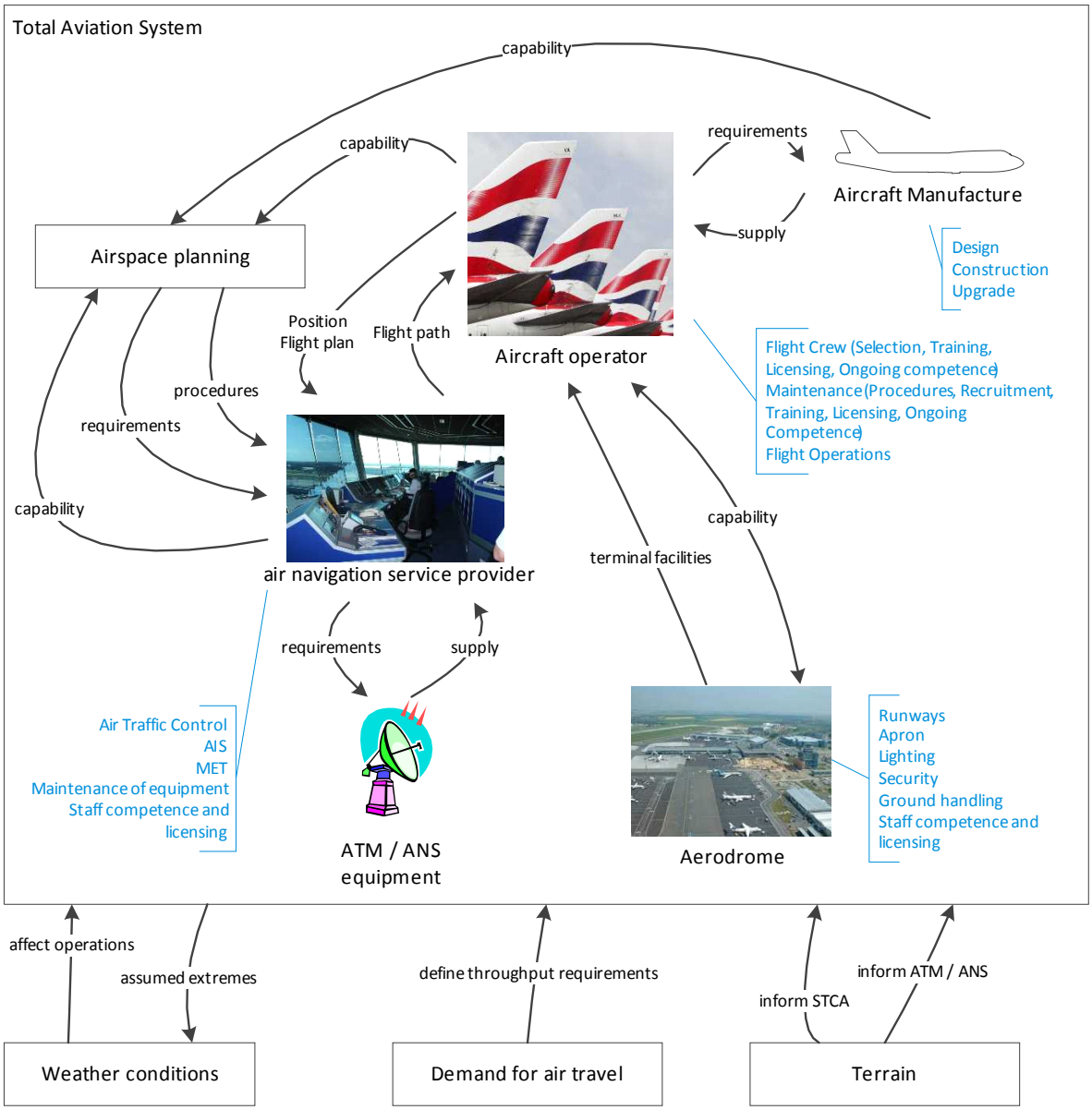

Figure 26: Functional breakdown of total aviation system (TAS)

In Figure 26, the TAS is subdivided as follows, and the interactions between these elements (and with the external environment) are shown:

- ATM / ANS equipment: this is the equipment used by the ANSP to provide the air navigation service.
- Air Navigation Service Provider (ANSP): 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: 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.

Appendix C 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' [12] for an overview of the notation and its rationale. A key to the symbols used in this document is given in Figure 27 and Figure 28 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.

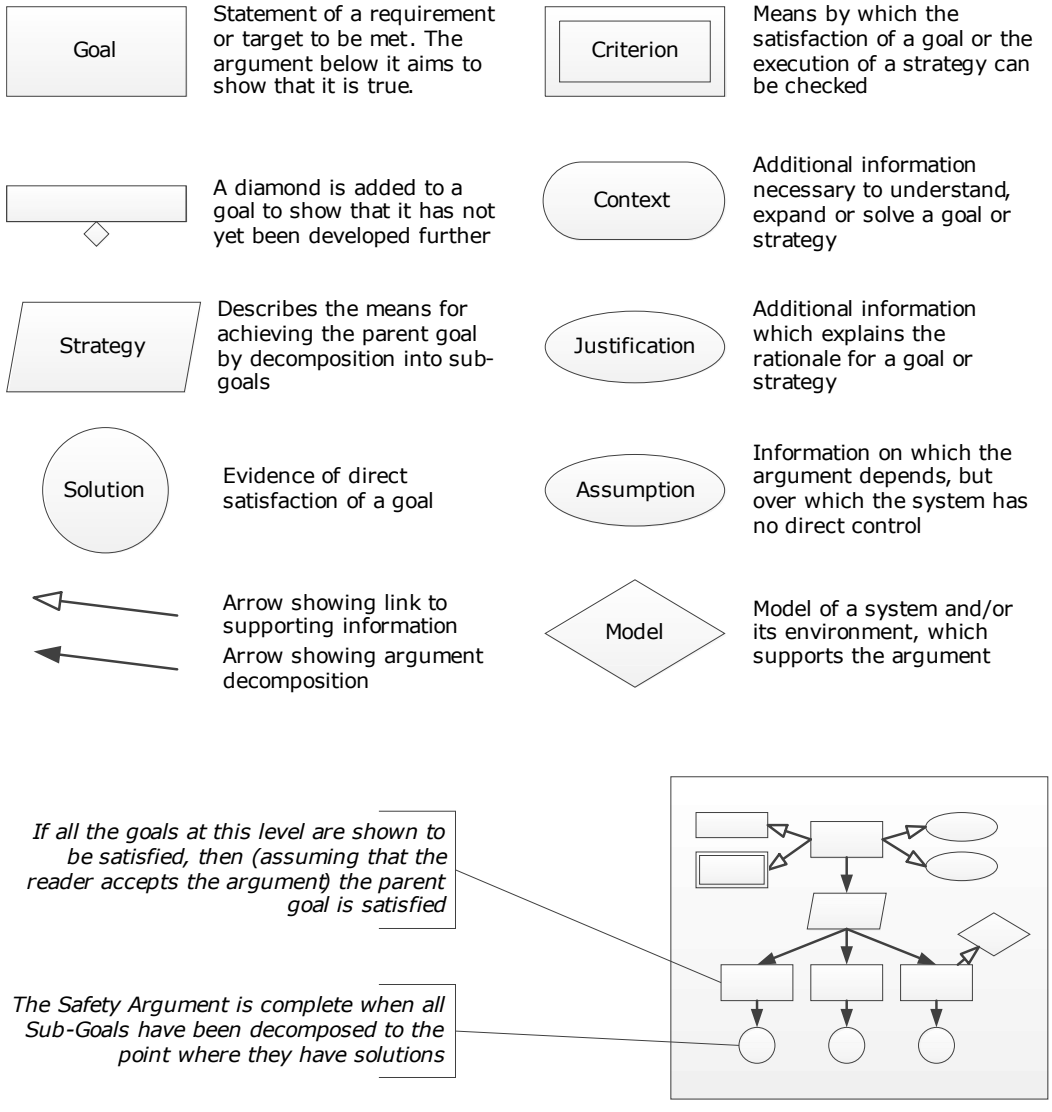

Figure 27: Key to basic GSN Symbols

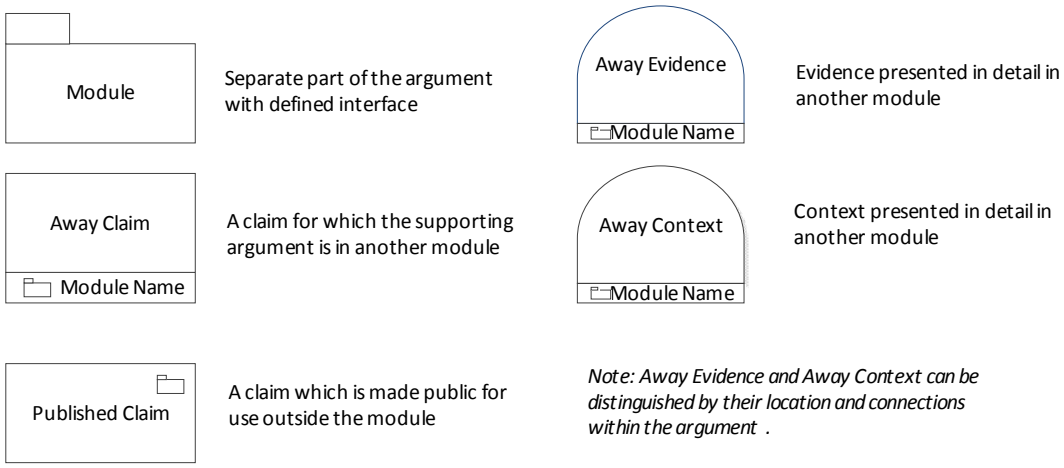

Figure 28: Key to GSN Symbols for Modular Arguments

Appendix D The Use of Safety Arguments in Industry

Safety arguments have been accepted across a range of industries for over 15 years as a means of enabling clear, concise and traceable arguments for safety assurance to be presented to regulators.

D.1 Development of Standards

ISO/IEC 15026 [41] introduces the concept of an Assurance Case as being the representation of a claim or claims and the support for these claims. The standard applies across the whole systems and software engineering lifecycle. An assurance case provides a multi-level structure of claims, sub-claims and connecting arguments that are ultimately based on evidence and assumptions that provide a reasoned, auditable argument supporting a claim – in essence a Logical Argument.

The Object Management Group® (OMG®) is an international, open membership, not-for-profit technology standards consortium, founded in 1989. The OMG have developed a Structured Assurance Case Metamodel (SACM) [42] which is a conceptual model for an assurance case structure. Part of the OMG SACM specification defines the Argumentation and Evidence Metamodels which facilitate projects by allowing them to effectively and succinctly communicate in a structured way how systems and services are meeting their assurance requirements. The SACM provides a modelling framework to allow users to express and exchange argument structures. Structured arguments comprise argument elements (primary claims) that are being asserted by the author for the argument, together with relationships that are asserted to hold between those nodes.

The Goal Structuring Notation (GSN) Standard [12] was developed by means of a consensus process involving GSN users from academia and industry between 2007 and 2011. GSN is a graphical notation that can be used to document explicitly the individual elements of any argument (claims, evidence and contextual information) and also the relationships that exist between those elements i.e. how claims are supported by other claims, and ultimately by evidence. Arguments documented using this notation can help provide assurance of critical properties of systems, services and organisations.

D.2 Previous Uses in Aviation

The *safety argument* approach has already been successfully applied to achieve *approval* for novel concepts in certain parts of the TAS, providing a degree of confidence of its suitability for use in the *approval* 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.

The EUROCONTROL Safety Case Development Manual (SCDM) [13] provides guidance on the approach to developing safety arguments in relation to the demonstration of the safety of a system or service within the aviation industry.

Past applications of the approach are numerous but include:

- the operation of military Unmanned Aerial Vehicles (UAVs) in non-segregated airspace [43]
- Reduced Vertical Separation Minima (RVSM) [44]

- the development of the Point Merge operational concept [45]
- the introduction of ACAS into European airspace [46]

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 *safety 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, see section 5.5.

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

- CAP670: Air Traffic Services Safety Requirements [37]
- CAP760: Guidance on the Conduct of Hazard Identification, Risk Assessment and the Production of Safety Cases [14]

D.3 Modular Arguments in Aviation and other Industries

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.

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 [47]
- Safety case architectures to complement a contract-based approach to designing safe systems [48]
- Safety Case Composition Using Contracts – Refinements based on Feedback from an Industrial Case Study [49]

Modularisation of arguments is already explicitly supported in some industries, as illustrated by the following examples:

- The rail industry (EN50129 [50]) 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.

- 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 [51], Clause 6.4.5.6) to avoid unnecessary replication of the activities.

- The notion of cross-acceptance 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 [52]. 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 [53]. There is currently a rulemaking task to develop this approach for Integrated Modular Avionics.