Establishing Safety Case Strategies for Mission Planning or Situational Awareness Systems

BJ Martin
Nova Systems
System Safety and Technical Airworthiness Competency Lead
Canberra, Australia
bj.martin@novasystems.com.au

Squadron Leader Derek W. Reinhardt
Royal Australian Air Force
Deputy Senior Design Engineer – Avionics C-130H/J
Air Lift System Program Office (ALSPO)
RAAF Richmond, New South Wales
Australia
derek.reinhardt@defence.gov.au

Abstract
Mission Planning and establishing Situational Awareness are important risk management strategies in complex and hazardous military aircraft operations.

Software based Mission Planning Systems (MPS) and Situational Awareness (SA) tools supporting operational decision making in circumstances that impact safety are now common place, and are becoming increasingly functional.

Operational approvals for such systems are typically based on satisfactory technical specification compliance and user trials with criteria of: effectiveness, workload reduction over manual methods, sufficiently intuitive interface, verified outputs for selected operational test cases; and qualified user workforce.

However, a conundrum remains for the structure of the system safety case argument, which would, in safety-related software theory, rely heavily on technical design assurances. The origin of many of the software tools forming part of a MPS is sometimes outside the environment where high integrity design assurance practices are common place. Often referred to in system safety literature as Software of Unknown Pedigree (SOUP). In this situation, the determination of a safety criticality / integrity level or hazard analysis activities do not typically drive system design requirements or design assurance activities. Therefore there are often substantial limitations in design development artefacts or other evidence that the software's integrity is likely to support the determination of safety criticality.

Instead, from consideration of instituted MPS and SA tool approvals processes, it may be construed that system Human Machine Interface (HMI) look-and-feel evaluation and user operational procedures are largely responsible for achieving adequate operational safety. Yet, rarely are effective human error or critical task analysis activities employed for these tools and functions, nor are workload assessments used to validate in-mission operators abilities to detect and correct errors before mishaps occur.

Examination of the limited literature or case studies identified of notable mission planning or situational awareness system related accidents, appears to weigh strongly towards user input or data related failures, and errors in correct system use due to incorrect initialisation or inadvertent reversion to default data values. These factors may be attributable to both technical and operational procedure design issues, although in some circumstances the causal factors have heavily favoured one over the other.

Where then, should the strength of argument and emphasis of safety case resources be invested for maximum safety return? What is an effective safety case assessment methodology for MPS or SA systems approvals?

This paper examines the current use of Mission Planning Systems, related accident history and causal factors, current regulatory requirements, and proposes a basis and methodology for architecting the safety case for MPS and SA systems.

Keywords: Mission Planning Systems, Electronic Flight Bags, Situational Awareness, Human Factors, HMI, Safety Case Argument.

1 Introduction
Safety certification of highly integrated technologies intended to perform a pro-safety service, and a bridging function between planning operations and actually conducting safe operations, can be, in the author’s experience, a vexed subject among safety engineers and operators alike. The technology in question does not directly control any hazardous energies, or directly cause mishap consequences when it fails; the usually drivers for safety integrity. However Endsley [EBJ03], Sandom [SaFo06], [San07] and Storey [StFa03] (among others) have published extensively on the relationships between SA, Information Systems, Data and safety. They argue that breakdowns in the functions these Mission Planning...

Copyright © 2011, Australian Computer Society, Inc. This paper appeared at the Australian System Safety Conference (ASSC 2011), held in Melbourne 25-27 May, 2011. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 133, Ed. Tony Cant. Reproduction for academic, not-for profit purposes permitted provided this text is included.
Systems and Situational Awareness (from here on ‘MPS/SA’) tools facilitate, can provide operators with misleading decision support data, which is frequently attributed as a major contributor to accidents.

The introduction of powerful computing and display tools for aviation MPS/SA functions has been gradually underway for nearly 20 years via the evolution to ‘data-hungry’ flight management systems. The first significant regulatory guidance on tools that support these functions was FAA AC120-76A [FAA03] for Electronic Flight Bags. [FAA03] proposes a traditional aerospace functional safety assessment approach, but only if the system was physically integrated into the cockpit. As with much emerging technology, the market was leading the regulators by several years, and there is much anecdotal evidence to support the notion that regulation is still playing catch-up. Thus, such technologies are yet to be heavily influenced by regulatory requirements, and operational approval authorities are left without a consistent understanding of the technical and operational risks presented by the products on the market (ie. there is no emergent and widely agreed technical certification basis for these MPS/SA tools).

Using primarily an aviation experience basis, this paper will examine the connection between MPS/SA tools and safety outcomes by reviewing noted accidents, incidents and analysis, exploring the contributors and comparing this to the relative emphasis and care given to these factors in development, introduction to service and regulation.

The authors have had involvement in the introduction to service and attempts to create safety case arguments for several MPS/SA and related critical data handling tools. The author’s observation is that while technical regulation is essential, much more work needs to be done for the integration of the technical evidence and arguments with the operational approval activities, in order to safely assess and operate these capable tools. This is vital to an overall safety argument architecture and safety case, and ultimately to ensure safety and benefits are realised by the employment of the MPS/SA tool.

The intent of the paper is to identify where the greatest returns for investment of effort are, in developing strong safety arguments for mission planning and situational awareness tools. It will discuss proposed methods of establishing arguments for safe operational release. The case is made for aviation applications, but read across for similar technologies by other safety critical industries will be possible. This paper has not, however, developed a honed methodology or cook-book for a complete safety assessment of MPS and SA tools. Peer validation of the strategies proposed here, further research and experience by application will be required.

The goal is to propose alternative pragmatic strategies for practice in this area, with well reasoned dependence on both technical and operational approval activities and evidence.

2 Background

To provide background to the human and technical factors examined by this paper, and their explicitness in the proposed safety argument, this section examines the MPS/SA tools, how they are intended to be used, how they have failed, and what this might imply for the safety case.

2.1 Mission Planning, Situational Awareness and Safety

The linkages and importance of the acts of planning and provision throughout of situational awareness, to flight safety, is largely accepted wisdom. Locally [CASA11] promotes it as follows - ‘planning is important because it constructs a four-dimensional picture of the flight in your mind’. That is, the act of planning builds a foundation for situational awareness. However, the degree of automation and complexity of the modern aircraft means that much of that situational awareness now resides in a machine. Human operators are instead presented with an emerging (but not necessarily intended) paradigm of merely following directions or monitoring automation.

Examples of this paradigm shift in the Australian Defence Force (ADF) context is reflected in the F-111 controlled flight into terrain accident in Malaysia in 1999 and the B707 crash near East Sale in 1991. Flawed mission planning and risk management were major contributors to these accidents, and the resulting investigations made recommendations for significant changes in regulating, and measurably improving, ADF operational airworthiness (safety).

While the paradigm of merely following the machines directions does offer some benefits in many circumstances, the operators ‘ignorance will become a problem if the machine stops’. Recent operational safety studies in Europe [Lea10] assessed helicopter safety incidents to identify major opportunities for improving safety. The studies identified four of the most insidious technical/operational hazards including: unexpected encounters with a degraded visual environment; getting into a “vortex ring state”; loss of tail rotor effectiveness; and static and dynamic rollover. A major contributing factor (or rather - weak defence mechanism) was inadequate mission planning, and the resulting shortfalls in situational awareness. These factors have become a focus of training leaflets being distributed by the European Helicopter Safety Implementation Team.

To further illustrate the criticality of effective mission planning, and the important role the MPS/SA tool has in this process, Section 2.3 of this paper will examine specific aviation accidents and incidents. It will indentify major contributing factors and will also illustrate challenges of new technology introduction, that brings both opportunities for improvement and new sources of hazards.

2.2 Use of MPS and SA tools

In aviation, trusted rules-of-thumb have always evolved built on lessons learnt, (eg. fuel reserves/flow rates,
Visual Flight /Instrument Flight Rules or Extended Twin Operations diversion limits, Lowest Safe Altitude and buffers for predicted performance/weather inaccuracies, etc.). Calculation tools have been used to support the planning function (eg. performance reference charts, whiz wheels, TOLD cards, and numerous home grown spreadsheets). This has been a natural response in order to reduce time and simple error prone, tedious and repetitive arithmetic. Sources and types of errors that might come from these methods and tools are generally well understood, the subject of explicit training, and are commonly the subject of flying supervision and pre-flight authorisation checks. However, like many aspects of modern life, the proliferation of automated tools is tending to drive complexity beyond the comprehension of most operational users. Undeniably though, the intent of adopting tools has been to improve safety and reduce incidence of hazardous simple errors; but it also creates the predicament of unknown, or at least under-appreciated, potential for error.

In aviation, modern MPS consist of software applications that allow maps, charts, weather, intelligence and aircraft performance data to be used in developing navigation solutions (e.g. routes, approaches, terminal procedures), communication settings, flight/mission calculations (fuel, leg times, etc), and other pertinent aircraft operational data. MPS may include visual software tools, optimised for specific aircraft roles, and automate the computations associated with aircraft specific flight/mission planning. Once the mission information has been generated, it is printed (e.g. kneeboards, strip charts), or alternatively written onto a data storage device (e.g. PCMCIA flash disk, or proprietary data transfer module) for transfer to aircraft systems (e.g. Flight Management System, navigation system, Electronic Flight Bag(EFB)). For the purposes of this paper, when MPS output data is combined with displays, and are used as a live updated decision support reference during operations - it is performing a Situational Awareness function. Some more recently developed MPS also include functions to transmit and receive flight/mission information via datalink, either at the commencement of a flight/mission, or as real time updates throughout a flight/mission. MPS may also be used for post-flight/mission debriefing and analysis. Note that while the FAA do not use the term MPS, preferring Electronic Flight Bags (EFBs) to describe these applications, the ADF use of the MPS term encompasses both flight planning and aeronautical database processing.

Aeronautical Data, which underpins many of the functions of the MPS tool, is provided to the ADF by the Royal Australian Air Force Aeronautical Information Service (RAAF AIS), and to civil operators by Air Services Australia. These agencies are charged with regional responsibilities for military and civil users, providing aeronautical data in electronic and paper form for planning, en-route reference and critical terminal procedures. How this data is integrated into an aircraft automatic flight management functions and used in operational procedures, governs the criticality of the mission planning system.

Similar parallels also exist in the maritime and land domains, such as the Electronic Charting Display Information Systems (ECDIS), integrated ship Navigation systems (eg. ECPINS), and Battlefield Command Support System (BCSS) to name a few.

In commercial maritime operations, similar applications in Electronic Charting Display Information Systems (ECDIS) and integrated ship Navigation systems (eg. ECPINS) have become more common place over the last 10-15 years for safety and economic reasons. Meanwhile the military has been adopting the same technology for arguably higher risk and more dynamic military operations planning for surface and now sub-surface applications. The drive for integrated automation and error reduction has also extended to join the digital dots between the source of the Navigation and operations planning data and the users. In Australia the Hydrographic Office, is responsible for charting and distribution of all Australian Territorial waters and additional areas of military interest. This service has transitioned in recent years from a historically evolved cartographic drafting service, reliant on evolved knowledge and craftsmanship, to a Digital Hydrographic Database importing survey data from a combination of survey tools (Laser Airbourne Depth Sounding, Survey ship digital soundings, and more manually collected then hand recorded into digital devices). This survey data is then collated and classified by the hydro data and charting specialists.

Irrespective of the domain the data is received, manipulated, created and transferred via various information systems and geospatial software tools linked together by an automated workflow, and tailored to meet international presentation standards as well as some client specific format requirements to suit particular MPS and SA tools used in operations. Tool developers of multiple application platforms will typically not be aware of the specific user operational context and criticality.

In the land operational environment, planning and control systems such as Battlefield Command Support System (BCSS) and other generic Battlefield Management Systems also rely on combinations of digital terrain data, scanned traditional topographic maps, live ‘blue-force’ tracking, mission planning overlays, dynamic intelligence data etc.

These non-aviation examples are described here merely to illustrate that the technical and conceptual issues to be discussed in this paper from an aviation perspective, will have applicability in other domains where safety and mission critical decisions will be made based on data presented and manipulated in integrated planning tools.

2.3 Examples of Accidents involving MPS and SA tools

A recent ATSB Report [ATSB11] catalogues and analyses 11 Australian and 20 International civil high capacity transport aircraft accidents and incidents over a 20 year period (Jan89-Jun09), where mission planning and data errors were involved. Three Examples will be reviewed briefly here to illustrate the scenarios:
2.3.1 Emirates A340 Melbourne - Mar2009

The following summary is based on the preliminary results of the ATSB’s ongoing investigation, released on 18 December 2009.

On 20 March 2009, the crew of an Airbus A340-541 aircraft arrived at the aircraft about 1 hour before the scheduled departure time. About 30 minutes later, they received the final loadsheet, with a Take Off Weight (TOW) of 362.9 tonnes. Shortly after, the first officer entered a TOW of 262.9 tonnes into the Airbus Less Paper Cockpit (LPC) electronic flight bag system. The first officer recorded the resultant figures on the flight plan and handed the LPC computer to the captain for cross-checking. The captain checked the take-off performance figures and entered the figures into the flight management and guidance system (FMGS). The captain’s figures were then cross-checked with the figures recorded by the first officer.

During the takeoff, the captain and first officer attempted to rotate the aircraft, but it did not respond. They tried again applying a greater nose-up command. The nose of the aircraft raised and the tail made contact with the runway. The aircraft did not begin to climb. The captain selected TO/GA thrust and the aircraft commenced a climb.

After establishing a positive climb gradient, the crew received a message from the on-board error system indicating a tailstrike. The crew notified Air Traffic Control (ATC) and advised that they would be returning to the departure airport. While reviewing the aircraft’s performance documentation in preparation for landing, the crew noticed that a TOW 100 tonnes less than the actual TOW had been inadvertently entered into the LPC, resulting in low V speeds. At no times during the process did the LPC or on-board systems challenge that the TOW might be incorrect.

2.3.2 MK Airlines B747

On 13 October 2004, a Boeing 747-244SF aircraft, registered 9G-MKJ, was planned to operate a multi-stage non-scheduled international cargo flight departing from Luxembourg, through Bradley and Halifax, Nova Scotia.

The aircraft was taxied to the runway and during the takeoff the aft fuselage momentarily contacted the runway. Several seconds later, the fuselage contacted the runway again with greater force. Contact with the runway continued to about 825 ft beyond the end of the runway, where the aircraft became airborne. The lower aft fuselage then struck an earth bank supporting the instrument landing system antenna and the tail separated from the aircraft. The rest of the aircraft continued forward until it struck terrain. The aircraft was destroyed by the impact forces and subsequent fire. All seven of the crew members received fatal injuries.

The following factors were identified throughout the subsequent investigation:

Flight data recorder comparison

The flight data recorder information for the take-off at Halifax was compared with the take-off at Bradley to identify any similarities. This comparison identified that the rotation speed and flap setting for both flights were about the same, however, at Bradley the aircraft reached rotate speed 13 seconds before that recorded for the Halifax takeoff, indicating a higher rate of acceleration. Furthermore, the initial pitch rate for the Bradley takeoff was 1.2 degrees per second and the aircraft climbed away about 4 seconds later, with the pitch angle increasing to 6 degrees. For the Halifax takeoff, the initial pitch rate was 2.2 degrees per second, with the aircraft lifting off near 10 degrees. This eventually increased to 14.5 degrees.

The take-off data for Halifax was identified as being nearly identical to that for the takeoff at Bradley, indicating that the Bradley TOW (239,783) kg was used to generate the performance data for Halifax. The calculated TOW for Halifax should have been 353,800 kg.

Boeing laptop tool (BLT)

In order to calculate the take-off performance data, landing performance data, and weight and balance information for a flight, the crew were required to use the Boeing Laptop Tool (BLT), which was located on the upper deck of the aircraft.

It was likely that the use of the wrong TOW came from the misuse or misunderstanding of how the BLT software functioned. When the BLT program was launched, the data for the previous flight would populate all of the fields, in this case, the data for Bradley. These fields would then need to be updated with the data for Halifax. If the user opened up the weight and balance page, and then returned to the take-off performance page, the TOW already in the system would automatically populate the planned weight on the take-off and performance page, which was 240,000 kg for Bradley. If the user was unaware of the software’s reversion feature or did not notice the change, and they selected the ‘calculate’ button, the resulting V speeds and thrust settings for the takeoff at Halifax would have been based on the data for Bradley. If these figures were written on the take-off data card with the correct TOW of 353,300 kg, it is likely that the error would have gone unnoticed.

Other factors identified

It was likely that an independent check of the take-off data card was not performed by the crew as required by the standard operating procedures (SOPs). The crew did not conduct a gross error check in accordance with the SOPs. The crew were at their lowest level of performance due to fatigue, which may have increased the probability of error when calculating the take-off performance parameters, and degraded their ability to detect the error. Crew fatigue and the dark take-off environment contributed to a loss of situational awareness. The airline did not provide formal training on the use of the BLT, nor did they have a proficiency program.
2.3.3 Southwest Flt 1248

After deciding it was safe to land in a snowstorm, the pilots of Southwest Airlines Flight 1248 overran the zone where the plane needed to touch down, resulting in a runway overrun. The result is that it skidded outside the airport and killed a 6-year-old boy who was a passenger in a proximate motor vehicle. The pilots needed at least 800 more feet of runway to avoid a collision, according to the National Transportation Safety Board (NTSB).

As they approached the airport the pilots and a Southwest dispatcher were confident a landing could be accomplished, despite contending with low visibility, a tailwind and reports of poor braking power on snowy Runway 31 Center. The pilots based their decision to land on the dispatcher's positive assessment, their piloting experience and flight data they entered into a cockpit computer. The onboard computer confirmed the difficult landing would be within the capability of the Boeing 737-700 and would conform to Southwest's procedures.

Flight crew used on-board laptop performance computer (OPC) to calculate expected landing performance. The OPC was programmed to assume that engine thrust reversers will be deployed on touchdown in its calculation of the stopping margin. The calculated stopping margin was acceptable to the aircrew. If the OPC did not use reverse thrust credit, it would have indicated that a safe landing on 31C was not possible. It's unclear whether the crew were aware of the assumptions in the OPC calculations. The NTSB now prohibits operators from using reverse thrust credit in landing performance calculations.

2.3.4 Other related accidents

The aviation accident record is focussed on heavy transport aircraft, where reliance on data is tightly coupled to hazardous and automated phases of flight. Other incidents related to similar causal factors and mishandling of automation include: Ryanair in 2006 [Lea06] a Controlled Flight Into Terrain (CFIT) accident was narrowly avoided after crew became fixated on reprogramming the automation via the Flight Management System (FMS) after the discovery of incorrect/outdated data for the airport they were approaching at Knock, Ireland. Another example of a more purely navigational data based accidents are the 1995 Cali B757, Flt 965 [Lad05] crash into mountainous terrain due to erroneous waypoint assignments by the crew when they lost situational awareness, were under time pressures, independent data validation sources had failed and they were subjected to arguably poor HMI design in the FMS.

2.4 Accident Reporting and Analysis

[ATSB11] states that it’s major findings corroborated previous findings from US NTSB and studies by Boeing and Airbus and were essentially that all identified incidents had causal factors associated with input errors, poor or non-existent gross error validation practices, time pressures, workload and/or poor coordination and communication within the crew or with external parties (ATC). These were seen by accident investigators as failures to create adequate procedural defences to error, and failure to recognise and react to abnormal performance aircraft indicators.

ADF Aviation Safety Spotlight Magazine 04/2010 [War10] picks-up this ATSB report and themes in “Deadly Data” and correlates to ADF experience with similar “safety factors” at play, in similar heavy transport operations. With the added complication of military transport involving more dynamic tasking but also subject to time pressures driven by operational imperatives and other safety factors, such as dangers to passengers in hostile zones if not airlifted etc.

The FAA have also reviewed specific incidents involving Electronic Flight Bags at [Chandra10] to identify some common threads and similar causal factors (ie training, familiarity, over trust without validating) but further drew out certain fundamental design features that were co-contributors to incidents and hazard scenarios. Particularly where the equipment was operated during the flight and involved crew workload/distractions from fundamentals of aviation because of legibility and manual manipulations, workload required to pan and zoom and allowing important context data to be missed.

Interestingly, this very issue was the primary contributor with a submarine grounding incident ([Per05], [Ham05]) where recently charted hazards had been updated on some resolution electronic charts but because the operator was using a lower resolution, the boat navigated into shoaling waters at high speed, despite active soundings cautioning the crew otherwise. Closer to home we read Newspaper stories of similar things happening regularly with over-reliance of drivers on SatNav directions in road vehicles.

[Sei11] and [FSF05] sites a current study collating NASA Aviation Safety Reporting System (ASRS) data where a general concern has been raised over reductions in pilots’ manual flying skills, possibly from an over reliance on automated systems, as well as an incomplete understanding of such computerized controls, planning tools and aircraft operating modes.

Avionics Magazine [Evans06], reviewed the celebrated example of the October 2004 MK Airlines incident in Canada. The planning errors were on an EFB tool, but are no different from those carried out on ground based Mission Planning Tools. The article criticises the design of the Boeing Laptop Tool (BLT) (and essentially it’s cousin products from other manufacturers that are no ‘smarter’) for not including design features that made modes and data manipulation actions more clearly understood by the crew, and checking for gross or nonsensical errors. It also has promoted HMI concept schemas that would help in this role. Finally the author advocates mandatory subjection of the laptop tools to the “same robust validation required of flight control software. At the end of the day, both are equally capable of killing”. This last statement in his editorial draws a much closer link between mission planning and catastrophe than is currently supported in regulations. The flaws were not system functional failures or erroneous behaviours. The failures identified were essentially in the requirements set, not having identified...
sufficient operational hazards and HMI challenges. So in the absence of systematic hazard analysis requirements in the regulations, success would have to depend on how operationally savvy the developers and testers were and whether they were testing for operator error potential and the validity of their requirements, both in normal and failure modes. This is not a common strength of the software development industry.

Several articles in aviation safety journals have focussed on the importance of recognising the training liability/burden of introducing MPS and SA tools in order to minimise human error and capitalise on the safety (and economic) benefits of such systems. This is a well supported focus from the analysis of causal factors above. But aren’t training and procedures the last refuge of the system safety scoundrel? In the MIL-STD-882C design mitigation order of precedence, we are supposed to deal with eliminating the hazard and providing safety features first.

2.5 Summary and Assessment of Contributing Factors to Accidents

All of these analyses support an obvious conclusion that better training and understanding is required in the automated systems and the vulnerabilities of the human and procedural interface. But is this sufficient or even practically maintainable given innate complexity and the pace of change in systems software upgrades? Is it a case of more sympathetic design to human responses and mental states?

Notably, none of the incidents and enquiries examined in the referenced reports, identified faults in the calculations performed by either planning devices or flight management computers. Although it has been suggested that these system human machine interface could have been designed more sympathetically and robustly, to identify and flag non-sensical or inconsistent planning data inputs and outputs. (ie. they could have been designed to add to the defences to assist error detection in critical tasks, if this operational criticality had been understood by the designers). It cannot be concluded that there were no software faults in those equipments used, but it is clear from the above analysis that tool faults (in requirements satisfaction) did not play a profound role in the accidents at hand.

In the absence of consistent application of airborne software standards and certification requirements for MPS design, what value do “robust validation” and operational testing provide? How reliant has current aviation EFBs been on a level of product quality that comes by default from existing trusted avionics suppliers, rather than on any demonstrated achievement of safety goals? As new software development sources and competition from cheaper providers come onto the market, how will regulators evaluate when the integrity of the design is insufficient?

The following section will now review what the current regulatory requirements are, and discuss how they are currently applied.

2.6 Current Civil and Military Regulatory Requirements and Application

The FAA’s AC120-76A was released in 2003 providing guidance for certification, airworthiness and operational approval processes of EFBs. It combined facets of existing regulation of airborne avionics device compliance and safety, with the results of sponsored Human Factors research by the Volpe National Transportation Systems Center [Cha03]. In brief, the requirements are graduated by classification of hardware (Classes 1, 2 and 3) and software (Types A, B and C) by features and functions with increasing approvals requirements and rigour as the system becomes more integrated into live operations as a reference and decision support tool. Physical cockpit integration is a primary indicator of safety criticality in this guidance material. As such the AC did not require functional hazard analysis (ala FARx.1309 system design and analysis requirements) of EFB systems unless they are a Class 3, permanently installed device. However, FAA inspector operational approvals include minimum requirements for training, currency and checking, operational and data update procedures, regardless of hardware or software classification, with increasing objective evidence requirements as the criticality increased. The minimum standards required for these operational procedures supporting EFBs installation, is very conceptual and subjective. The accident record collated by the ATSB and other cited examples, seem to indicate that consistent application of the principles and intent in AC120-76A are not yet common place or fully effective.

The ADF technical airworthiness regulator, DGTA, has considered its approach to EFBs and MPS for a number of years while dealing with the more gradual integration into modern military aircraft. The military regulator has also had to consider the broader context of integration of MPS and data input/output into military theatre operations planning systems. Preliminary guidance had been available for design requirements and Technical Regulatory compliance since 2004 and in 2008 DGTA (including one of the authors of this paper) published a Notice of Proposed Rule Making (NPRM) for EFBs [DGTA08] and later for the broader scope of MPS in Dec’09 [DGTA09]. The EFB certification guidance considered AC120-76A applicable in most cases of common mission planning system functions, however the discriminator of criticality of function of mission planning systems was to be dictated by the level of automation and risk associated with the operational roles of user aircraft. This, then identified a need for further guidance of how to establish a basis for judgement of criticality. The MPS NPRM directed that aeronautical data and it’s intended use was this discriminator. Depending on whether the aircraft and crew would rely solely on the data correctness for safety critical functions or decisions, would dictate the criticality of the MPS functions of generating, manipulating and transferring this data. The systems carrying out these functions – software, hardware and human – inherited this criticality and responsibility for integrity. (This concept of data
criticality will be explained further in Section 3.1 of this paper.)

On the operational regulatory side, however, there are limited ADF requirements or guidance published. The technical regulations outline some operational safety management considerations, but operational approvals are not equivalently or explicitly regulated as they would be for the FAA requirements on EFBs. Military Aviation Regulation 6 [ADF09] is currently interpreted to consider EFBs and MPS as classes of Aviation Support Systems, and thereby requires classification, requiring a form of technical approval and an Operating Permit. The requirements of a basis for an Operating Permit are less explicit in terms of assessment, procedures, currency and approval requirements and not specific to mission planning systems.

CASA’s published guidance appears in a very recent publication of an Airworthiness Bulletin [CASAA10] which would essentially indicate that the FAA view, requirements and comprehensive approach should prevail.

Alas (and anecdotally) - a recent unattributable presentation on an approach being taken by a low cost airline to ‘paperless cockpits’, was witnessed at an Aviation symposium in Australia. This presentation underscored the naïve but perhaps understandable approach possible in this immature area of the power of emerging technologies. Automation may be sought as a business solution in order to simply reduce operating costs, without foreseeing a safety implication of the negative sides. In the subject presentation, an in-house developed set of calculation tools, hosted on a commercial mobile computing device, were developed to reduce perceived overhead, improve planning speed and on-time departures via flexibility for re-planning in the cockpit, and reduce take-off settings to minimum margins. According to the presenter, acceptance of the new system was to be based on a judgement of minimum negative feedback from operating crews in trials, and having achieved local CAR35 signatory approval for carriage of the device based on no physical interaction with the aircraft systems. The fact that this approach had reached implementation trial stage in a public transport carrier, indicates a worrying absence of understood technical and operational approval requirements based on functional criticality, or at least a basic lack of awareness of those regulations that should apply.

In summary, it seems that regulations and guidance in the aviation sector with regards MPS approvals is currently available but fluid and dispersed. It is also clear that the technical approvals basis is more rigid, and primarily driven by considerations of physical interface to the operating platforms rather than consideration of the functional interface. The exception to this is where data criticality is being proposed as a discriminator, such as the ADF draft design requirements for MPS. Even in this case, the data criticality discrimination, is being used as a mechanism to drive software assurance requirements (not yet a substantial contributor to the accident record) and not human factors assessments or regulated operational approvals.

2.7 State of practice for certification of MPS/SA tools

Despite the recent emergence of the certification requirements discussed above, many MPS/SA tools in use haven't typically been developed to these frameworks, or have ignored normal airbourne system certification requirements. Further the incidents and accidents record suggest that at this time the frameworks are yet to be totally effective.

MPS tools are not, in totality, considered as aircraft software, and thus they may not be subject to normal aircraft software certification requirements. For example, as described in the section 2.5, the FAA approach to Electronic Flight Bags only prescribes full software assurance certification requirements for specific types and functions of tools (e.g. Type C tools), while limiting software certification requirements for other types of tools (e.g. Type A and B tools). This approach is pragmatic, but it also means a wide range of tools are outside traditional aircraft software certification requirements. Hence, it is relatively common practice of not subjecting such tools to safety and design assurance practices commensurate with airborne software.

So why does a certification authority take this approach? One reason is that many of these tools offer significant improvements to pilot planning efficiency and situational awareness (hence safety dividends). The other is that for many of these tools, the worst credible hazard may only be as severe as Minor when appropriately incorporated into normal cockpit procedural practices. However, when it comes to the application of such policy, developers of these tools may not always understand these underlying assumptions behind the policy. Hence it is common place to see developers promoting products for which very limited certification evidence exists. This practice exists because often some developers are ignorant as to the certification requirements, and are focussed on the perceived benefits of the tool. For example the laptop, netbook and iPad evolution the proliferation of software applications on these platforms has lead these developers to explore applications across a wide range of domains outside the conventional IT domain. Further, the developers are likely not to have undertaken to fully assess the hazards associated with the use of the tool, or have blindly made underlying assumptions that there will usually be a human operator in the loop, and this will provide sufficient mitigation to all classes of errors, faults and failures under all circumstances. Yet, rarely is there any evidence of such an assessment. Very recently news items were posted at [Jep11] that an iPad application has received EFB certification by the FAA at Class1 level, with aspirations for Class 2.

In the military environment, explicit regulation of certification requirements for such tools has lagged somewhat. The ADF’s NPRM on Flight and Mission Planning Systems [DGTAA09] is still one of the first military authorities to publicly issue certification guidance for MPS and EFB tools. While this guidance is derived from the FAA approach mentioned earlier, it is adapted for military operational circumstances, and
heavily technically focussed, as technical and operational regulation is a separate activity in the ADF.

The most common MPS in use with the ADF are the Portable Flight Planning System (PFPS) and the Joint Mission Planning System (JMPS). The dependability of these products relies substantially on the retrospective Verification and Validation undertaken by the USAF and USN respectively, rather than the prescription of development assurance practices. One of the author’s on this paper has direct experience of studying the qualification and release processes, where (for example) the USAF assigned test verification agency is manned with several hundred personnel dedicated to testing, verification, validation and support of the system PFPS for every operational platform in USAF service (including FMS export aircraft types). This agency undertakes a substantial V&V, including regression program for each PFPS build delivered. While this approach does not mirror the application of software assurance principles of recognised assurance standards such the coupling of ARP4754/61 with RTCA/DO-178B, Def(Aust) 5679, and Defence Standard 00-55 (now obsolete), it does contribute evidence to the safety case with respect to confidence in PFPS's behaviours.

Several legacy ADF MPS acquisitions have also relied heavily on one-time-only ADF conducted Verification and Validation to provide some assurance against hazardous behaviours (e.g. the Mission Data Preparation Equipment software for the now retired F-111 aircraft). For example, when MDPE was being accepted by the RAAF, one of the authors of this paper was personally involved in the V&V of the MDPE software consisting of many thousands of V&V cases of the software, including functional, robustness, and crew procedural requirements. This V&V was undertaken over the period of 4-6 months. The fault density in earlier versions of MDPE was relatively high, however, the V&V effort certainly contributed to the subsequent resolution of many of these issues that might have provided an opportunity for a hazard to the crew. Again, this didn't constitute standard software assurance practices that would see this evidence generated during the development of the product, nor is this approach being currently advocated by ADF regulation. ADF’s resources to achieve this are not the same as in the mid-90’s when much of this work was undertaken. Nonetheless it did contribute to the case for acceptance/employment of MDPE. Still, for other ADF aircraft, assurance of mission planning systems have been almost completely overlooked in the development stages. In either case, the V&V effort possible today in Australia pales by several orders of magnitude to that nominally provided to PFPS and JMPS as a matter of course.

To summarise - assurance practices have had very limited application to MPS/SA tools developed to date. In some cases V&V has been used a means of shoring up the design evidence shortfall, but this is becoming less practical in the current Defence funding climate, and not practical in a competitive airline environment. Therefore, achieving assurance of MPS/SA tools requires a more holistic approach.

3 Case for Data and Design Integrity

Section 2.3 has summarised several hazardous aircraft circumstances related to MPS tools. Of note, most of these were largely the result of operator errors and misunderstandings of the tool's results, however the integrity of the underlying data, and of the tool itself is an essential input to safety. For example, there is evidence of tools providing invalid results that were interpreted to be valid by human operations. This is potentially a shortfall by the human in interpreting this information, but also a requirements validity issue with the tool if invalid data can be interpreted as valid data. There is also evidence of tools invalidly using stale data in calculations. Further still, there is evidence that the workload impost on human operators working with these tools (particularly those used in flight) resulting from unexpected or unintelligible behaviours of these tools is also a factor. All of these circumstances are evidence of potential shortfalls in the integrity of the respective tools, albeit they are concerned more with requirements validity than with latent faults in the specified implementation. For the purposes of this paper, integrity is a qualitative term used to infer the degree of confidence that the software's behaviours are valid in both normal and failure modes of the software and that the behaviours that may impact safety satisfy an explicit and valid requirement for that behaviour. Therefore, software integrity is not just the isolated application of software assurance practices, but the application of software assurance in the context of allocation/derivation of requirements for the software from safety analysis of the system, the software and the operational context.

Further to these more obvious factors, there is another factor to consider which doesn’t get tied directly back to the individual circumstances surrounding just the way these tools have failed in practice. There is a broader question of when the tool may be found to be a contributor to an accident, what will the investigation recommendations (hence opportunities for litigation) be targeted at. The authors' view is that accident investigations will often make recommendations for explicit behaviours of the tool. For example the findings made against the Boeing Laptop Tool mentioned in section 2.3 relate to clear annunciation of stale or default data to the operator to avoid misinterpretation for valid date. There are also recommendation relating to interface regarding display of units, etc; all of which are functional requirements for the MPS tools.

The other forms of recommendation may be less explicit but become more apparent if litigation is pursued. For example, if an unassured tool calculated an invalid result which was misinterpreted by the human operator; and it could be demonstrable that the application of industry benchmarks and recognised aviation software practices for this tool would have prevented the issue at reasonable cost then would the developer of the tool be held liable? There are few cases to date where this argument has been explicitly tested. The authors' reason that the comparison of what’s been done regarding tool assurance, versus what could have been done at ‘reasonable’ cost, would
Certainly fall against developers who had adopted the former approach.

To a great extent, the regulatory requirements outlined in Section 2.5 provide some benchmarking of what forms a suitable basis of comparison, but in the ADF context, these are yet to be widely adopted.

So in light of these circumstances, and the assertions this paper has made about the behaviours of certification authorities and developers alike, and how these may be traced to the incidents of the previous section of this paper; what is the case for prescribing data and design integrity requirements onto MPS tools? This paper proposes that the case should be centred around two key issues.

The first is that the errors, faults or failures of the tool should not present an unreasonable burden on the human operators in having to dedicate crew resources to detecting any errors, faults or failures. This is because the very purpose of the tool is to reduce workload in planning and situational awareness, and not to add workload burden to these activities. Ideally there should be no errors, faults or failures (but for pragmatic reasons, absolute assurance is not achievable). However, since errors, faults and failures are almost inevitably present, the focus should be to use assurance practices to limit the presence of these errors, faults and failures to within a tolerable operational burden (i.e. the satisfaction of requirements of the tool should be assured to a known confidence). Safety and software assurance practices currently offer the only recognised approach to presenting an argument that qualifies the degree of confidence in the absence of errors, faults and failures. In it’s frequent absence in the MPS/SA context, however, the ‘assurance deficit’ must be assessed for acceptability of risk by some means. This is where the linkage into the human factors evaluation elements become explicit (refer to Section 4).

The second is that the suitability of the tool’s behaviours should be explicitly treated, through the introduction/confirmation of assured product behavioural requirements for those circumstances which would increase the likelihood of the crew invalidly interpreting a result from the tool (i.e. the tool should be designed to actively minimise crew error and prevent hazardous circumstances). An example of these sorts of circumstances might be a small error in flight performance information that leads to a runway length error in marginal operating circumstances without crew knowledge. Another may be a small positional error that leads to invalid situation awareness regarding navigation, particularly when engaged in tactical low-level flying. Both of these circumstances could be mitigated through the introduction of safety requirements (reasonability checks, cross checks, warnings, etc) to assist in their mitigation. To make the safety case successful, developers should have to argue that the tool has been designed not to reinforce or contribute to a decision process which would lead to hazardous circumstances. For example, if a tool can allow default or previous data to be automatically imported into the tool's fields to expedite repetitive entry tasks, then the possibility of this data being invalid should be explicitly treated. This may only be achieved via the introduction of specific design safety requirements on MPS tools. If considered for developmental tools, where the opportunity to inject design requirements still exists, this will be relatively straightforward. However for legacy tools, it is rarely possible to retrospectively inject design requirements to introduce such behaviours to the tool. Instead then, such legacy tool circumstances will drive a need for human operational evaluation to assess the real affects of the absence of these features, and determine their tolerability. Until design regulation leads the technology, this latter situation will carry the greater burden of the safety argument.

In light of these two key arguments, the following sub-sections outline the case for data and design integrity for MPS tools. The paper has been constructed around both data and design integrity because the results of these tools inevitably depend on both the behaviours of the tool, as well as the data that is the input to these tools.

### 3.1 Aeronautical Data Integrity

Aeronautical data integrity is the degree to which confidence can be placed in the precision and accuracy of the supplied data. In circumstances where aeronautical data, or the calculations being made from it, are used to support phases of flight where errors in that data may be hazardous, then clearly the integrity of the data is paramount.

While it is possible to argue, that aeronautical data integrity uncertainty, and the hazards arising from its use, may be overcome by detection and workarounds by human operators, this approach is problematic. Humans are relatively good at detecting significant gross errors, provided the relevant cues are provided, and the basis of comparison from which the error is detected via comparison of similar information (e.g. similar types of displays and units). Unfortunately, humans are less adept at detecting subtle errors between information. The challenges with aeronautical data are that there is so much of it (i.e. the aeronautical data bases are usually big and complex), and this prevent it being obvious to operators as the how good all that data is, unless appropriate controls have been placed on how the data has been generated, manipulated and managed.

Under what circumstances then would it be reasonable for a human to detect an error within aeronautical data? This will be dependent on the extent to which the MPS tool manipulates the data or derives other values from it. Further, other data types may be used directly in the tool output, but the detect-ability of their validity (and any margins of tolerability of accuracy and consistent will depend on how the data is used and correlated by the pilot to other situation awareness cues.) Section 4 of this paper deals with these aspects of the operational assessment. Any data which fits beyond the reasonable detect-ability and handling by operators should therefore be subject to some form of dependability assurance. This seems a pragmatic approach, but when subject to operational hazard assessment, much data used for the more challenging operations, will always end up requiring a
level of assurance, as it just isn’t possible for a human operator to provide appropriate workarounds to potential shortfalls. One approach to aeronautical data assurance might be along the lines of the FAA approach for data integrity:

- **RTCA/DO-200** – Standards for Processing Aeronautical Data describes the requirements for the processing of aeronautical data including tool qualification requirements.
- **RTCA/DO-201A** – Industry Requirements for Aeronautical Information specifies the aeronautical data elements required by the aviation industry and a standard for the accuracy, resolution, and integrity of the associated values.

The FAA approach is consistent with ICAO practices and is one of the more mature approaches available. Of course, given that in many cases the regulation lags innovation, this is not necessarily testament that the FAA approach is the best approach. Note also that the FAA approach to Aeronautical Data Integrity is not a one size fits all approach, and it scales the degree to which confidence is required along similar lines to the Design Assurance Level approach of software standards such as RTCA/DO-178B.

Alternatively the ADF has developed an approach that adapts the FAA approach to the military specific context (refer to the NPRM for AAP7001.054 Section 2 Chapter 24). The ADF approach encourages an even more product focussed assessment of the data in the context of the tool and end application to establish the impacts of invalid data, and this drives data integrity requirements.

In many cases operational evaluations will lead to the conclusion that the data needs to be dependable, and thus the data integrity requirements will be required anyway for the bulk of data being used to support MPS tool functions and the flight operations dependant on them.

### 3.2 Software Safety and Assurance

Software safety and assurance are the means by which software is developed that meets safety objectives. It usually involves a complementary suite of analysis and verification evidence which seeks to show two key outcomes: requirements validity, and requirements satisfaction. The following sub-paragraphs examine these two concepts in further detail, and explain their relevance to MPS tools.

#### 3.2.1 Requirements Validity

Requirements validity addresses the question: does the software have the right behaviours? It is about ensuring that both the normal functional behaviours of the software, and also the failure behaviours are compatible with the intended safety objectives. While this is a fairly abstract concept, it provides some important pointers for the types of behaviours that have to be considered when developing software requirements, which will ultimately determine the acceptable behaviours for the software.

So what are the acceptable behaviours for an MPS tool? There are several ways to achieve this. One practice is to start with a system with known behaviours and empirically evaluate the suitability of each of these behaviours in the operational context. While in many respects this provides an very effective evaluation of these behaviours, it is costly and time consuming. The alternative approach is to establish a set of behaviours early in the development and subject them analytically (or by targeted operational evaluations) to establish their suitability and completeness. If this second approach is adopted, then the result should be requirements that unambiguously define the functions associated with each piece of information presented to the user, and why this behaviour is appropriate under all scenarios that this information may be used. This will provide the requirements for the normal functional behaviours of the system.

Further to the normal functional behaviours of the software, additional behaviours of the software should be defined. These are the behaviours of the software to deal with circumstances involving errors, faults and failures that might invalidate the normal functional behaviours of the MPS software detailed above. Behaviours dealing with errors, faults and failure normally require two properties, one of which is the means by which the error, fault or failure will be detected, and the other is the means by which it will be handled. In some cases the handling of the error, fault or failure, may be by defining a requirement for a behaviour at the human machine interface that alerts the human operator to the error, fault or failure.

Therefore a key aspect of identifying and analysing the behaviours of an MPS is to ensure that the evidence provides good coverage of both these normal and failure circumstances. One way of achieving this would be to undertake analysis that considers:

- each resultant piece of information presented to the human operator, or prepared for automated transfer to an aircraft (such as to a flight management system),
- each phase of flight, or operational scenario in which this data might be used (e.g. power settings supporting take-off, a GNSS based landing approach, etc.)
- the credible effects (including worst) of the information being invalid in that flight scenario,
- whether the invalid information would be human detectable, and what the impact on pilot work-flow would be as a result of the error (see section 4)
- how the MPS tool could either prevent, or detect and handle the applicable error, fault or failure

In the ADF, this approach as described by AAP7001.054 Section 2 Chapter 24, has had some limited application to a couple of applications, eg PPFS/JMPS, Super Hornet, MRH90 GMMS. In each case, the understanding developed as to what the data each system provided and how it was used, permitted an extremely pragmatic approach to fielding these systems (albeit retrospectively) to be pursued.
However, despite the intentions of the aforementioned analysis, when evaluated, many MPS will have behaviours that aren’t appropriate under certain circumstances. If this is the case, then the effects, human detect-ability and impacts of human work-flow are vitally important. An example of this is some of the limitations promulgated by the ADF on the use of PPFS and JEPS in flight to support more challenging navigation functions, in the absence of full operational evaluation of potential workarounds to the shortfalls. Section 4 describes how these should be evaluated and how it might be established that these collective impacts are tolerable.

3.2.2 Requirements Satisfaction

Having developed a set of requirements that are asserted to provide a set of behaviours compatible with safety objectives, requirements satisfaction deals with the implementation of these requirements such that the required behaviours are implemented in the software product. The key goal is to ensure that in implementing these required behaviours, that unacceptable errors are not introduced that would lead to a hazard to safety, or violate an assumption about treatments to identified hazards.

For aircraft software, requirements satisfaction is normally achieved via the application of the software levels within software assurance standards such as RTCA/DO-178B. While there are arguments in the literature about the effectiveness of such standards, the purpose of this paper shall not be to re-examine these arguments. Instead, this paper will assume that whether it be the framework of a software assurance standard, or the alternative framework provided by an argument and evidence based approach, both are means of providing evidence of requirements satisfaction. For the purpose of simplicity here though, these concepts will be referred to as the application of a software assurance standard.

For new ADF specific developments, ADF contracts (or equivalent instruments) should include requirements for software assurance to the degree required when the MPS tool employment is holistically considered.

However, the earlier paragraphs in this section discussed the limitations to the applications of software assurance standards to all classes of MPS and EFB tools per both the FAA and ADF policy. In these the application of software assurance practices could be retrospective at best; and most likely perceived as not cost effective. Therefore, in the absence of prescription of a software assurance standard, how does the FAA and ADF policies assure requirements satisfaction for these tools. In short, this is where the interactions between the degree of design and data integrity and the human factors elements come into play. There is no blanket answer on whether all the circumstances where safety and assurance practices haven’t been employed would result in acceptable tool solutions. Section 5 presents a safety case argument strategy that deals with this conundrum.

4 Case for Human Interface Safety

Mission Planning Systems and SA tools, by their very definition, are an extension of the Human operators thinking and computing space. The operator exports data into the tools in order to simplify the computation of derived parameters and assist in creating and representing mental models back to the operator in preparation for use. This data may then also become “off-line” during operations, but be used for real-time decision making. It follows that the interface and presentations through which this information transfers is critical to crew manipulating data correctly and gaining a correct understanding of the results.

Dr. Carl Sandom, an independent consultant in the arena of safety-critical software based systems and human factors, maintains a thesis of the interdependence of human safety functions and system safety functions in information systems. “Human as Hazard or Human as Hero” is his catch phrase. In [San06] and others, he portrays the interaction of real-time and near real-time information systems, operations and safety outcomes in the following model:

![Human Interface Safety Model](image)

Fig 1 attempts to convey the global picture of influence on accident sequences where, for safety related information systems, the operational sphere has almost as much influence on the outcome as the core system. The consistent concern is that design and safety assessment analyses focus too heavily on system functions and integrity, without the commensurate analysis to support assertions of human function reliability and the factors that would affect it.

It is reasonably clear from the above referenced accident research, that erroneous human interactions with planning and automated flight management systems, is a vulnerable link in the aviation safety chain. Appropriately, the FAA’s AC120-76A approvals requirements guidance, devotes significant proportion attention to Human Factors in design and implementation.
requirements. This content was largely taken from the FAA sponsored research published by the Volpe Center in 2000, which has since been significantly superceded and embellished at [Cha03], and recognises that although EFBs may increase efficiency and safety of operations, they “could have negative side effects if not implemented correctly. For example, increasing workload and head-down time, and distract crews from higher priority tasks”.

In 2010 Volpe commissioned Chandra et al again [ChaK10] to review safety incidents involving EFBs and this report validated that workload, data entry errors and crew attention fixation issues as major contributors, all revolving around the way the EFBs were implemented and trained.

The authors believe that several conclusions are reasonable from the accident analysis, and when correlated with extant Human Factors accepted studies collated in [Kel85], [EBJ03], [San06] and others from various industries:

- Humans are traditionally poor at the role of passively and continuously monitoring automation for long periods of time;
- There is a natural bias towards trusting automation even when external cues and training would indicate otherwise; and
- Increases in automation are reducing aircrews fundamental skills to deal with failure scenarios.

The last point has also been discussed recently in aviation safety journals and commentary such as [FSF05] and [Sei11]. Further the accident record seems clear that operational procedural norms and training have not yet been adequately ‘tuned’ to new vulnerabilities introduced by planning information systems and flight management automation.

In other words – managing failure at the MPS/SA human machine interface is an undeniable and critical link to achieving operational safety.

4.1 Assessing Safety Hazards from Human Factors

What does this tell us about hazard elimination or acceptable risk assessment? Safety management will reside in a combination of design features, highly integrated planning and operational procedures, and effective training. Where the design stage is evolutionary or all together independent of implementation, a greater weight of responsibility for safety falls on the design of operational integration.

In either case, a more comprehensive hazard analysis activity is required. An analyst first will have to identify and assess the criticality of where the most hazardous elements (considering both human tasks and system functions) exist, then have to identify what features or procedures would facilitate error detection and correction, in various phases of operation, considering workload and other factors affecting probability of error reduction in order to make risks acceptable.

[SaF06] describes one such approach to address these issues by defining a framework for identifying human safety and system safety requirements through a design phase using established Critical Task Analysis and Human Error Analysis methodologies. [SaF06] describes how to assess an existing COTS/MOTS product’s safety for a new application, in the cases where safety requirements have not been explicitly articulated and tested, and safety or user error reduction features not documented in any analysable form. Data criticality analysis and software integrity requirements can be decided by the methodologies discussed earlier, but it is reasonable to assume (and is the well-worn experience of the authors) that if these considerations were not part of the original design, then objective evidence of design assurance is difficult to achieve retrospectively. However, Critical Task Analysis (CTA) and Human Engineering Assessment (HEA) activities are still valid are certainly able to be applied retrospectively. Human error detection and correction (or handling) is the last line of defence against hazardous system faults and errors created by human functions.

In the realm of aviation electronic flight bag functions, the Volpe design assessment guides are clearly a good starting point for hazard identification sources. Deficiencies against these proposed design features are immediate candidates for hazardous interactions. A data criticality analysis will then focus identification of the more critical functions as related to the target application and user scenarios. In order to complete the data criticality analysis, a complete understanding of operational intent and user operational workload profiles will be required through engagement with platform qualified operational representative/s and reference to an authorised Statement of Operating Intent, which would bound the problem space. Moreover, in the case of pre-existing or non-safety assured systems, it is possible and may be necessary or even essential (in the absence of sufficient human factors analytical resources) for extensive Operational Evaluation to pre-date formal ‘service release’ in order to fully appreciate the critical tasks and human error zones, training needs and procedural defence targets.

4.2 Mitigating Human Interface Safety Deficiencies

Fundamentally and inevitably the human functions will be both “hero and hazard”. No two humans are alike, so it will not always be possible to transfer assumptions from one human operator to another. Where analytical safety assessment or operational evaluation identify vulnerabilities and deficiencies in design, the choices are limited. From [DGTA09], the following approaches are outlined to overcome an identified technical shortfall:

a. Design Assurance.

b. Detection and Handling Mechanisms.

c. Operational Limitations.

d. Independent Verification.

e. Risk Retention.
Experience in the ADF has shown that options ‘a’ and ‘b’ are typically a last resort of project managers, because they have a substantial cost and schedule burden, and usually only pursued after clear demonstration that a combination of options ‘c’, ‘d’, and ‘e’ are unworkable, and the risk is not tolerable.

The most difficult challenge in this step, is typically the lack of sufficient substantive human factors analysis and/or operational evaluation results to make convincing unacceptable safety arguments. In the absence of thorough analysis and data, only opinion and credibility are available to ensure sufficient safety is provided. All of which may be flawed or skewed.

Insufficient benchmarks or other objective measures currently exist. Unfortunately, the apparent convenience and lack solid counter evidence of the efficacy of these mitigations, means they are accepted. Thus, with these emerging technologies and accompanying hazards, the task of demonstrating adequate safety of operational risk management measures, will often be harder or easier depending on specifically relevant experience levels of the operational regulator staff.

4.3 Procedures, Training and Workaround Options

Standardised procedures, supervision and checking are essential and common defences for safety related human functions. These are normally designed around intuitively critical steps, or aspects that were complex and prone to human error, or applied in response to experience of failure. With emerging technologies being charged with automating previously human functions and adding more complex simultaneous activities, intuition is no longer enough and experience is not available.

For complex systems, as identified above, a combination of specific analysis methodologies and structured operational test and evaluation periods are likely to be required to in order to develop appropriately targeted procedural defences.

In particular, human procedures that are intended to compensate for design integrity shortfalls must be based on study of the practicalities of detection and correction of each critical potential error in function or data handling, with due consideration given to a reasonable workload in envisaged phases of flight including degraded visibility or weather conditions or predicted system failure scenarios. In order to detect error, crew must have ready reference “truth data” that is regularly being checked. For example, monitoring aircraft performance against plans is typically well supported by constantly scanned instrumentation and back-ups. However, navigation cross referencing requires visual meteorological conditions and independent sources of position data from the integrated flight displays. It is generally understood as difficult to for operators to detect subtle errors in digital aeronautical data. This should best be achieved via automated and qualified data integrity control. If this isn’t in place, then operational procedures to limit the effects of invalid data to minor failure conditions only - some analysis would be required to work out how each data element is used. Eg. think about how to detect if a runway threshold is in the wrong position for a GPS based landing in Instrument Meteorological Conditions - crew would have to be correlating differences between the GPS solution and the older navigation beacons. A high workload impost and a busy stage of flight.

5 Safety Case Argument Strategy for MPS & SA tools

Based on the two cases presented above (for software and human factors), this paper proposes a top level argument strategy for MPS tools that might form part of the overall safety case argument for such tools. Note that additional factors such as tradeoffs between operational risk, capability and safety during non-peace time operations have not been covered within the argument strategy presented in this section.

As the case for human operator responses relies on the impact of the behaviours of the MPS tool, and the case for suitability of MPS behaviours relies on the existence of features of the MPS tool to avoid potential sources of human error. The top level argument of the proposed strategy focuses on the duality of the interactions between the human operation and the MPS tool. Figure 2 presents a Goal Structuring Notation (GSN) representation of the overall strategy proposed by this paper.

![Figure 2: Top Level Argument Strategy](image)

The argument focuses on a systematic evaluation of the interactions between the human and the MPS tool, to ensure that each of these interactions are acceptable. Two main arguments make up examining these interactions, the human factors element (G_Human) and the MPS tool (G_MPS) element. This deliberate breakdown ensures that neither human factors evaluations, nor design and data integrity form a biased role in the argument, and that both have equal intended precedence, and cannot achieve the requisite outcomes in isolation of each other. The following subsections now examine the two key sides of the argument strategy.

5.1 Human Factors Elements

The main thrust of the human factors elements of the argument strategy is to ensure that the human operator responses to the MPS tool’s behaviours, with an emphasis on those behaviours which might be considered erroneous or invalid, are appropriate. The strategy for this argument is to systematically examine, both in isolation and
collectively, each identified MPS tool behaviour and to
determine if the associated human responses are
appropriate.

The multiple instantiation dot on the link to the goal
G_Human_X indicates that this goal will require
instantiation for each MPS tool behaviour. So how
should each MPS tool behaviour be established (per
C_Human)? This is one element of the argument where
there is an implicit dependency between the human
and MPS tool sides of the argument. Where design and data
assurance practices have been employed on the MPS tool
side the argument, then the behaviours of the MPS tool
documented in requirements, verification and validation
evidence naturally provide a basis from which to infer
MPS tool behaviours. Where the MPS tool argument is
substantially weaker, then this puts an imperative on the
human evaluation program to focus more analytically on
potential behaviours of the MPS tool (such as via some
structured software safety analysis, functional analysis,
etc), to draw meaningful conclusions about having been
systematic about MPS tool behaviours in the human
evaluation.

To determine if each MPS tool behaviour is acceptable,
both the detect-ability of the MPS tool behaviour is
considered, along with the human handling response to resolve. The argument also makes explicit the phase of
flight, as the impact of MPS tool behaviours will almost
certainly always vary depending on the phase of flight in
which the information is required or used. Figure 3
presents a GSN representation of this part of the argument.

Figure 4: Detect-ability, Handle-ability and Workload
For the purposes of this paper, the remaining lower level
parts of the argument are left undeveloped and
uninstantiated. Below this level will always be evidence
and solution dependent, and it is not the purpose of this
paper to provide anything more than a generic argument
strategy template.
5.2 MPS Tool Elements

The main thrust of the MPS tool elements of the argument strategy is to ensure that a set of suitable MPS tool behaviours is provided to avoid those circumstances which would result in potentially hazardous human operator circumstances. This argument establishes that within the workload and crewing procedures established for the crew to safety operate the aircraft, no behaviours of the MPS tool should violate these in such a way that leads to hazards. The strategy for this element of the argument is to systematically\(^2\) evaluate what might constitute potentially hazardous human operator circumstances resulting from MPS tool outputs, and then examine ways the MPS tool might offer additional behaviours to prevent or avoid these circumstances. Figure 5 presents a GSN representation of this part of the argument.

---

\(^2\) The intention of requiring ‘systematic’ evaluation is to ensure that the operational context is defined, and the means of undertaking the evaluation provides a measure of coverage of the extent (and confidence) in the circumstances identified by the evaluation.

---

*Figure 5: MPS Tool Elements*

The key point here is that this element of the argument does not focus on MPS tool errors or failures, instead it should be examining how the MPS tools outputs (even when correct) affect the human workload and procedures. As we discovered with the human factors side of the argument, this here introduces the mirrored implicit relationship between the sides of the argument. To completely understand the human operators’ workload and procedural implications for the outputs of the MPS tool, then strong linkages into the human engineering program results will be required. This is reflected by the context C_MPS, and the relationship it infers. Figure 6 shows a GSN representation of a strategy of how C_MPS might be presented, such that the linkages into both the human engineering program and the safety program are made explicit.

---

*Figure 6: Evaluating Human Operator Circumstances*

The multiple instantiated dot on the link to G_MPS.1 in Figure 5 requires that each potentially hazardous human operator circumstance gets considered, so that the MPS tool outputs that affect each of these circumstances get systematically reasoned about. There is no need to introduce any non-interference criteria between the MPS tool outputs at this level of the argument, as this will be made more explicit at lower levels of the argument associated with MPS tool behaviour implementation, or requirements satisfaction in software assurance parlance.

For each MPS tool behaviour that might provide avoidance of the hazardous human operator circumstances identified above, there are two key elements that provide assurance of that behaviour:

- Safety and design assurance of the MPS tool behaviour itself; and
- The integrity (accuracy, precisions, etc) of the underlying data used by the MPS tool behaviour.

Figure 7 shows the elements of the argument that introduces the role of software safety, software assurance and data integrity.

---

*Figure 7: Software Assurance and Data Integrity*
For the purposes of this paper, the remaining lower level parts of the argument are left undeveloped and uninstantiated. Below this level will always be evidence and solution dependent, and it is not the purpose of this paper to provide anything more than a generic argument strategy template. However, the application of recognised safety standards (ARP4754, DefStan 00-56, etc), software assurance standards (RTCA/DO-178B, Def(Aust)S679, DefStan 00-55 (superseded)), and data integrity standards (RTCA/DO-200A and RTCA/DO-201) would provide an appropriate instantiation of these lower level goals. While it is recognised that many MPS tools are developed outside such frameworks, and the retrospective application of such standards may be problematic, these goals provide the linkages to those circumstances when an absence of assurance (in the context of the overall human operator interaction with the MPS tools), would likely be intolerable.

6 Summary

This paper has considered the challenge of creating a successful safety argument for implementing the emerging technology of sophisticated and integrated mission planning systems and situational awareness tools into safety critical operations, such as aviation.

Usage has evolved over the last 10-15 years in commercial and military aviation and the accident record now provides sufficiently valid data to identify the consistently contributing factors to catastrophic outcomes.

The authors have assessed how well matched current regulatory guidance is to these contributing factors and how it is currently being applied to product development and implementation. The paper will then consider the relative contribution of software design and data integrity on balance with human interface design and operational assessment and training, for mission planning systems operational safety. There are challenges for assessing and mitigating the hazards posed by each.

Finally the paper proposes the elements of a safety case argument structure that may be used to achieve approval for use of mission planning and situational awareness systems in safety critical applications.

7 Conclusions

Current certification and operational approvals requirements for aviation mission planning systems (including EFBs) and situational awareness tools are arguably not sufficient. Potential accidents due to MPS or SA tool causes are not mitigated to an equivalent risk level as for other hazardous and catastrophic potential aircraft systems. System design standards do not exist and valuable lessons learnt are therefore not able to feed into an improving and safer design basis to be consistently applied. Operator interface with these systems is the most vulnerable link in the accident causal chain, which needs to be supported by more robust designs and critical task analysis leading most specifically to tailored operational procedures. Analysis should also be supported by conducting thorough operational evaluation in order to develop targeted training, currency requirements and aeronautical data management processes.

In the interim, individual applications for certification and implementation of mission planning systems should be required to present a more sound and substantiated safety argument, fulfilling goals of a balanced treatment of design integrity and human factors elements, where each arm supports the assurance deficits of the other - as proposed in this paper.

It is hoped that continued interest in the subject for aviation safety and it’s potential extrapolation to other safety critical industries, and decision support information systems, will result in further research and validation. Most fruitfully, a cut-set of certification design requirements (or System Safety Requirements) could be identified. These would include HMI and safety features, as well as design pedigree and software assurance that will influence the state of the art offered to the market.

8 References


[AA03] AC120-76A, “Guidelines for the Certification, Airworthiness and Operational Approval of Electronic Flight Bag Computing Devices”

[RTCA] DO-200, “Standards for Processing Aeronautical Data”


[DGTA08] NPRM01/08, “Electronic Flight Bags”, Preliminary DRAFT AAP7001.054, Sect2.Ch22


JAA Leaflet No. 36 Use of EFBs


[Cha03] Human Factors Considerations in the Design and Evaluation of EFBs, v2, DOT-VNTSC-FAA-03-07, September 2003, Volpe centre


