Active Safety for Aviation♣

B. Kirk, Prof. I. Schagaev**, Dr. T. Wittig***, A. Kintis****, T. Kaegi*****, F.Friedrich******

**Robinson Associates, Weavers House, Friday Street, Painswick, GL6 6QJ, UK*
***London Metropolitan University, 166-220 Holloway Road, London, N7 8DB, UK*
****Euro Telematik AG, Riedwig 5, Ulm, D-89081, GERMANY*
***** SPIRIT S.A., Syngrou Avenue 377, Paleo Faliro, 175-64, Athens, GREECE*
****** ETH Zürich, Clausiusstrasse 59, CH-8092 Zürich, SWITZERLAND*

1. Abstract

This paper presents the main results of the EC co-funded, FP6 Research Project called 'ONBASS' (On-Board Active System Safety) [1]. The aim of this project is to develop the 'Principle of Active System Safety' (PASS) for aviation, with general aviation being the primary field of application. Rather than simply recording data during flight in order to support post-crash analysis (as is mostly the case today), ONBASS proposes to analyse the currently available data, as well as 'historic' aircraft data accumulated from previous flights, in real time, during the flight and continuously react on it with the aim of accident prevention.

2. Introduction

Traditionally, depending on the approach adopted, safety enhancement measures in the Transport Industry can be categorised into two types: 'active' and 'passive'. For example, in the Road Transport Industry, passive measures for enhancing safety would be the seatbelts, the airbags, steel reinforced car frames etc. On the other hand, active measures would be the modern lane control systems, proximity warning systems, as well as the ABS (Antilock Brake System) and EBD (Electronic Brake Distribution) systems.

In other words, passive safety measures have been around in the Road Transport Industry for a number of years now, whereas active safety measures are only recently becoming more and more common. Still the progress achieved by the Road Transport Industry in the direction of active safety measures has been quite impressive and swift. In contrast, in the Aviation Industry - apart perhaps from Military Aviation – active safety measures are basically non-existent.

The ONBASS project aims to address this 'gap' by developing both the theoretical foundation and associated models, as well as the primary functionalities and elemental architecture that will enable the development of a system which will actively, and in real time, prognose, analyse and address/mitigate any flight hazards on-board and consequently aid in accident prevention. The initial field of application chosen by the ONBASS consortium for this study was GA (General Aviation). The reason for this decision was that it is the least complex field (in terms of systems and parameters available on-board) allowing thus an easier demonstration of the basic associated principles, while it is also the area where the ONBASS consortium partners' main expertise lies. On the other hand, access to the necessary technical data was considered much more straight-forward than the highly competitive commercial aviation field or the classified military aviation field.

♣ This work was supported financially by the EC Project Framework 6, Thematic Priority: 'Aeronautics and Space', under Contract No. AST4-CT-2004-516045. The name of the associated project is 'ONBASS' (On-Board Active System Safety).

3. Theoretical Foundation & Model [2]

To establish the basis for achieving the challenging targets of ONBASS, the consortium partners began by studying and analysing the typical General Aviation aircraft, its systems, components and the functions that these are responsible for carrying-out, with the goal to devise the optimum technical solution which will empower the full deployment of the ONBASS system capabilities.

In following, the theoretical foundation of the 'Principle for Active System Safety' was established, starting from the 'Model of Active System Safety for Aviation' (MASSA). Sensors on-board the object (aircraft), continuously provide values for the flight data parameters. These represent, often indirectly, the condition of the object, its elements, the sub-elements thereof, and so on and so on. Each particular condition of the object might in some way be related to the parameters collected, (i.e. the values may directly or indirectly indicate a fault in some component/element/system). In turn, several consecutive flight data records examined together may provide evidence of a trend (i.e. some fault/malfunction developing). The implementation of PASS assumes that many events that would reduce aircraft safety can be avoided by continuously monitoring and analysing the condition of an aircraft in real time. These events could be predicted and acted upon by analysing the flight data available in-flight, in conjunction with 'historically' stored data (from previous flights) of the same aircraft. The overall model (MASSA) used to analyse this flight data during any given flight comprises of an object (in this case, the aircraft), its elements (i.e. the major 'divisions' of the aircraft, namely the structure, the engine(s) and the systems), the functional models of these elements, the set of operational flight modes, the flight data available in real-time, the predicates of the object and/or its elements, the elements' states, a dependency matrix defined with respect to the object's elements and a recovery matrix. The structural organisation of MASSA is illustrated in Figure1 below:

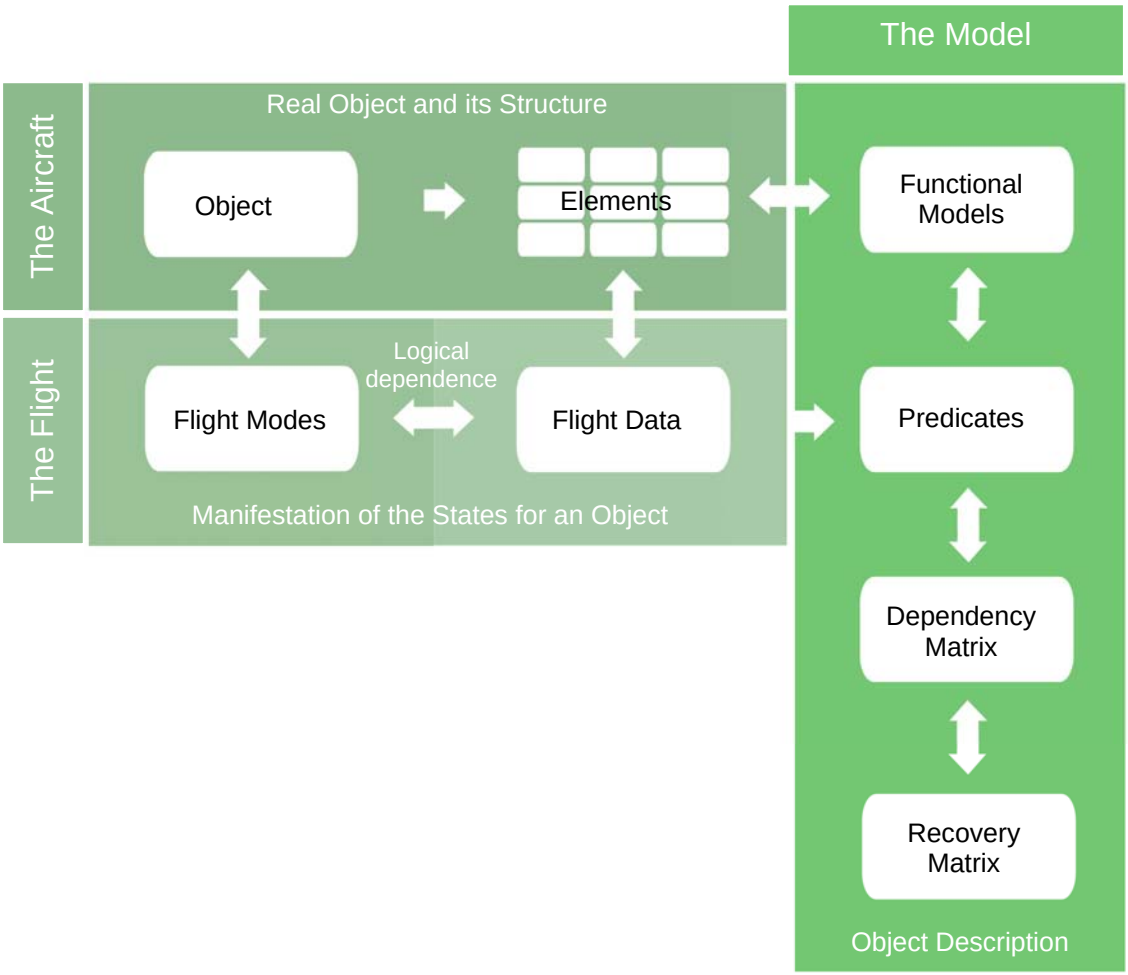


Figure 1: Structure of the Model of Active System Safety for Aviation

The main (Level-1) aircraft elements as previously discussed are the aircraft structure, engine(s) and systems. Sub-elements of these (Level-2) would be the wings, generators, fuel system, landing gear, control system etc. In Figure 1 such an object and its elements are represented in the top left corner. They exist in the real world and their conditions, as far as they are known, are reflected in recorded flight data. Note that the condition of one element might be reflected/recorded in various records of flight data, i.e. there is not necessarily a one-to-one mapping between elements and the flight data recorded.

To monitor the behaviour of an aircraft in terms of safety, a set of models for each individual element are used and these can be based on functional, probabilistic, threshold or other techniques such as those illustrated in Figure 2 below:

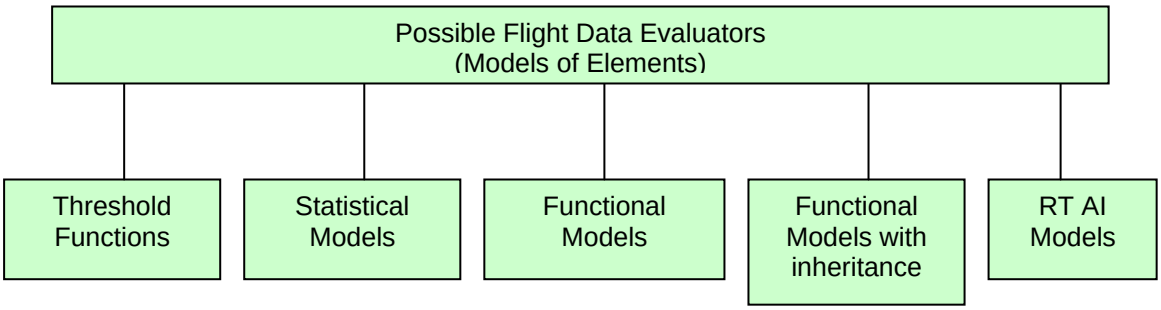


Figure 2: Possible element modeling techniques

The condition of the elements forms a vector of predicates, the so-called 'syndrome' of an object. These syndromes are snapshots that describe the condition of the aircraft in terms of the faults of its elements. There is an undetermined -in many cases- dependence between the various aircraft elements in terms of faults and the chain of events/sequence of possible malfunctions (see Figure 3).

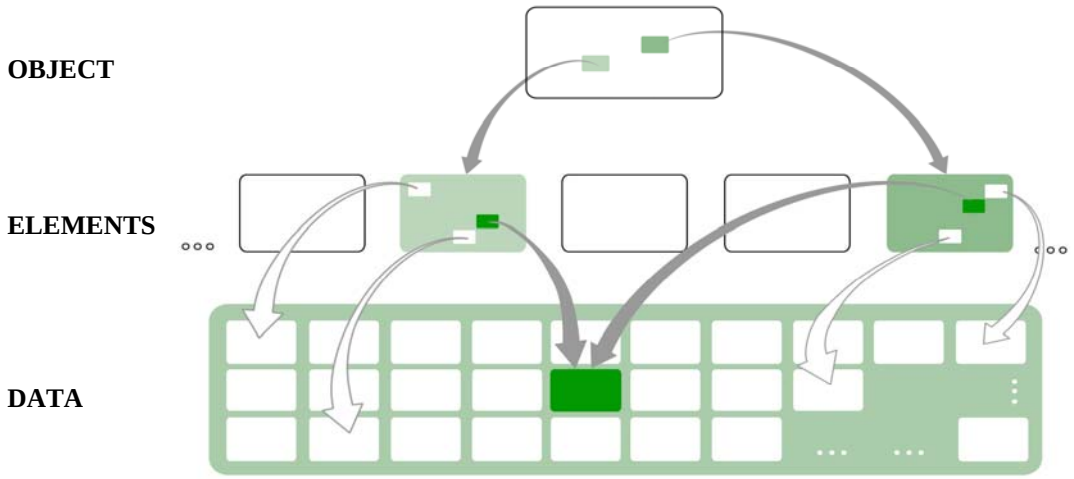


Figure 3: Dependencies of data and elements

This dependence may further vary greatly as a factor of the flight mode the aircraft is currently in, such as takeoff, climb, cruise and landing. The associated 'consequences' will also vary depending on the operational flight mode considered as the faulty behaviour of each element has potentially a different severity, for example, on the ground, than in the air. The underlying dependence relationship (both in terms of the possible 'development'/'evolvement' of a malfunction as well as in terms of the severity of the possible impact) is reflected in the matrix of mutual dependence, the so called 'Dependency Matrix' which uses a directed graph to represent this information. The Dependency Matrix describes the possible dependencies (relations) between the elements, sub-elements and components thereof of the aircraft, in terms of fault influence and propagation. The simplest version of this matrix is a square matrix that has *n* columns and rows and describes possible dependencies of *n* elements of the object

(aircraft). There are several alternatives for implementing the dependency matrix in the MASSA, using various mathematical techniques such as the Boolean matrix, undirected graph, directed graph and probabilistic graph. An example of a probabilistic graph and a probabilistic matrix are provided in figures 4 and 5 that follow:

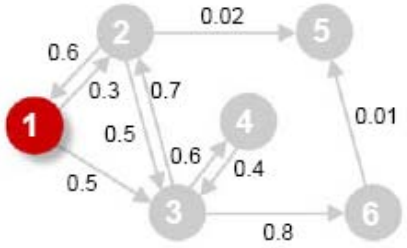


Figure 4: Probabilistic Graph Dependency Matrix representation

	1	2	3	4	5	6	7	8	9	10	11
1		P_{12}				P_{16}			P_{19}		
2	P_{21}		P_{23}		P_{25}		P_{27}				
3		P_{32}									$P_{3,\ 11}$
4					P_{45}			P_{48}			
5		P_{52}		P_{54}						$P_{5,10}$	
6	P_{61}						P_{67}	P_{68}	P_{69}		
7		P_{72}				P_{76}					
8				P_{84}		P_{86}					
9	P_{91}					P_{96}					
10					$P_{10,\ 5}$						$P_{10,\ 11}$
11			$P_{11,\ 3}$							$P_{11,\ 10}$	

Figure 5: Probabilistic Dependency Matrix representation

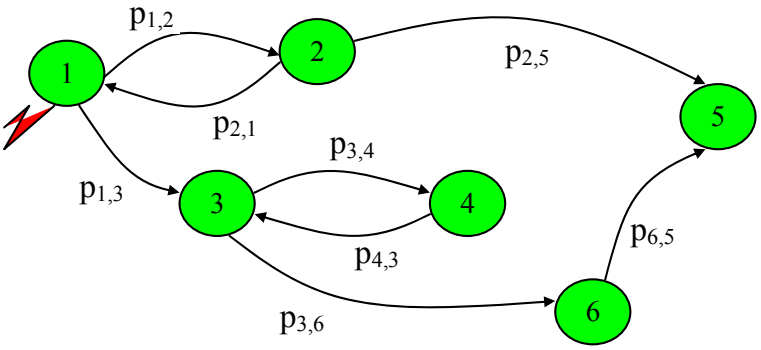
In the graph of figure 4, every matrix element r_{ij} is defined according to the rule: $r_{ij} = 1$ when an object element e_i functionally relates 100% to another elementary object e_j. The dependencies between elements in terms of safety can be described in terms of probabilities. These probabilities of possible transitions between the i-th and j-th elements of the Matrix, i.e. the probabilistic dependence of two elements (the i-th and j-th element) in terms of fault dependence might be substantially different, i.e. a fault of one element probably causes (induces) a fault in the other, but not necessarily vice-versa. In other words, for elements i-th and j-th, two probabilities P_{ji} and P_{ij} are defined and if P_{ji} is a probability that element j induces a fault in element i, then generally $P_{ij} \neq P_{ji}$. Their interactions in terms of induction of faults can thereby be represented. Regular statistical analysis can be used to tune a model of element dependencies by updating it with newly discovered dependencies and possibly excluding existing ones that have become obsolete. For an aircraft, this means that statistical analysis to upgrade the dependency matrix should be performed autonomously after each flight to take into account the changes in the condition of MASSA elements. Note that tuning of the MASSA is performed only post-

flight to avoid inconsistencies, and potential safety hazards, in real time in-flight data processing within a single flight.

The alternative ways to react to a hazardous condition that the object (aircraft) is subjected to, arising from the specific condition of all the object's elements, are defined in the Recovery Matrix. The Recovery Matrix maps one-for-one to the Dependency Matrix, (i.e. if the Dependency Matrix has dimensions *nxn*, the Recovery matrix will also be a *nxn* matrix). The use of the Dependancy Matrix makes it possible to analyse and define "what are we going to do" when a particular hazardous situation occurs. The analysis of this matrix provides a powerful tool to define the possible consequences of faults that appear in the aircraft. Two processes are defined on the matrix presented above as part of this analysis:

1. The possible consequences of a fault are investigated and defined.
2. The locus or loci, of faults, i.e. the element most likely to be source of the malfunction is identified.

The first process is all about making a prognosis about a possible flow/chain of events and the associated severity. It is initiated by information derived from flight data analysis regarding existing systematic discrepancies which are intrinsic to the aircraft's design, construction and operation. The associated process is developed as an algorithm of diagnosis and prognosis. The second process implements the investigation as to the instigating element, for the manifestation of the discrepancy, i.e. it provides an answer to the question where does the 'root cause' lie. This is made possible by following a procedure referred to by the consortium as 'reverse tracing'. During this procedure the 'path' of greatest probabilities is followed in a reverse order until the 'root cause' element is determined. The associated 'instruments' used are illustrated in the following figure and include the Dependency matrix and a directed probabilities graph:



$$D^{(6)}= \begin{pmatrix} p_{1,1}, p_{1,2}, p_{1,3}, p_{1,4}, p_{1,5}, p_{1,6} \\ p_{2,1}, p_{2,2}, p_{2,3}, p_{2,4}, p_{2,5}, p_{2,6} \\ p_{3,1}, p_{3,2}, p_{3,3}, p_{3,4}, p_{3,5}, p_{3,6} \\ p_{4,1}, p_{4,2}, p_{4,3}, p_{4,4}, p_{4,5}, p_{4,6} \\ p_{5,1}, p_{5,2}, p_{5,3}, p_{5,4}, p_{5,5}, p_{5,6} \\ p_{6,1}, p_{6,2}, p_{6,3}, p_{6,4}, p_{6,5}, p_{6,6} \end{pmatrix} = \begin{pmatrix} 0.00, 0.60, 0.00, 0.50, 0.00, 0.00 \\ 0.60, 0.00, 0.50, 0.00, 0.02, 0.00 \\ 0.00, 0.70, 0.00, 0.60, 0.00, 0.80 \\ 0.00, 0.00, 0.40, 0.00, 0.00, 0.00 \\ 0.00, 0.00, 0.00, 0.00, 0.00, 0.00 \\ 0.00, 0.00, 0.00, 0.00, 0.01, 0.00 \end{pmatrix}$$

Figure 6: An example of the 'reverse tracing' process

As a result of this analysis of the Dependency Matrix, the Recovery Matrix is defined. The Recovery Matrix identifies a set of possible reactions to the detected or suspected faults/malfunctions. Having defined the 'root cause' of a fault or malfunction and the associated severity (as described above), it is quite straight forward to identify the most optimum ways to address the situation. Each cell of the Recovery Matrix thus contains two values: the addresses of the ONBASS core application components that should be activated (when the MASSA determines possible success of recovery) and when exactly

this should be done (i.e. the appropriate timing). It should be noted that MASSA assumes that there is a possibility for non-absolute recovery. The process illustrated in the following diagram defines how the Recovery Matrix is populated initially, how it is used, as well as how it is thereafter updated.

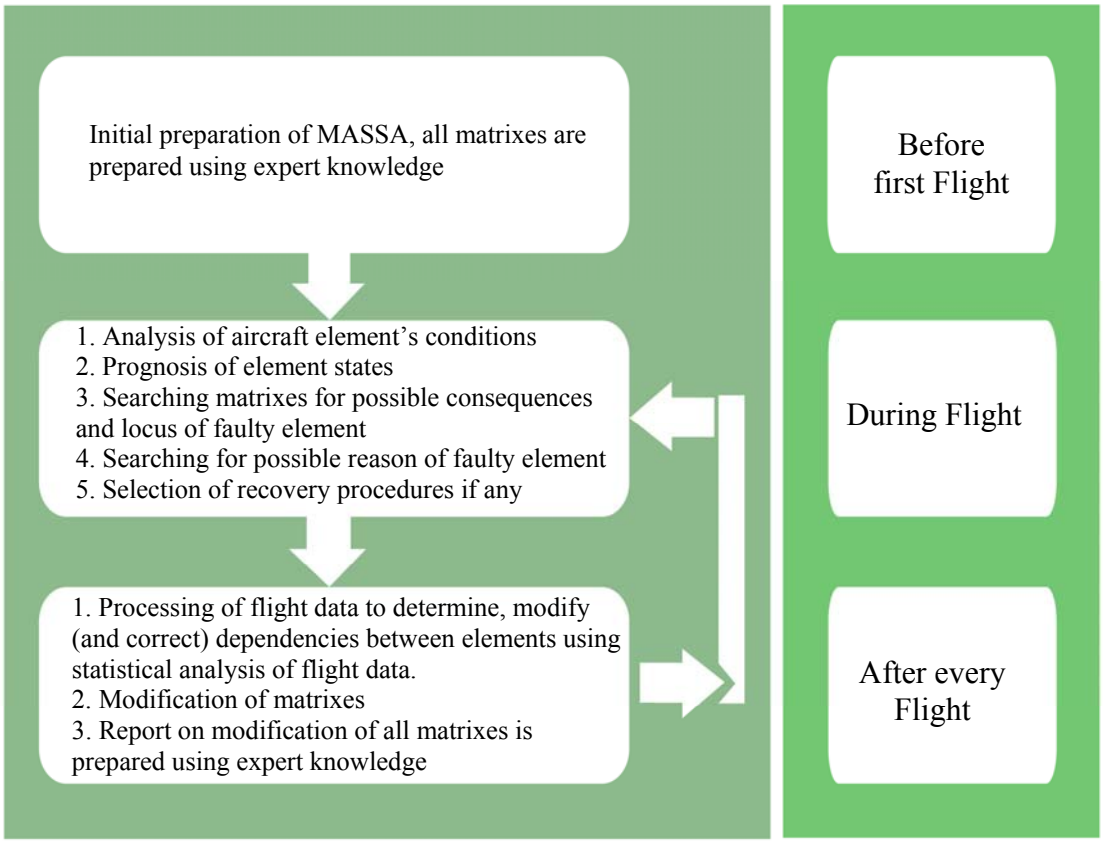


Figure 7: Creating, using and updating MASSA

To summarise, the MASSA implementation performs three functions during its principally different phases: on the ground (before the first flight), on-board (during a flight) and after each flight. Note that safety/technical experts for the particular aircraft in question prepare the initial values of the MASSA matrices. All subsequent 'tuning' is processed post-flight using accumulated flight data and the existing matrices by the system itself. During any flight there is a high quality prognosis of the current and projected aircraft conditions using the MASSA and more specifically the Dependency Matrix.

4. The System

Having established the theoretical foundation for ONBASS, the consortium moved in the direction of establishing the relevant system architecture and design. The result of this exercise is illustrated in the following figure:

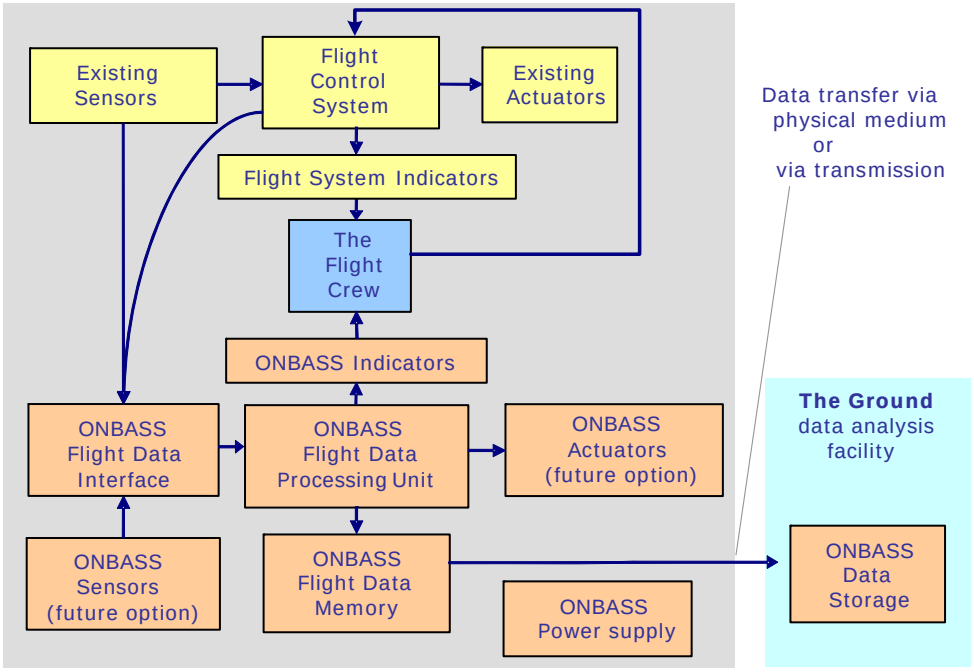


Figure 8: ONBASS System Architecture

The ONBASS system (illustrated above) comprises of two main parts: the ONBASS core unit and the ONBASS HMI (Human-Machine Interface) unit. In turn, the ONBASS core unit comprises of the Flight Data Interface (FDI), the Flight Data Processing Unit (FDPU), the Flight Data Memory (FDM) and an independent power supply. A typical installation in a GA cockpit for evaluation purposes would look something like that illustrated in the following figure:

Figure 9: ONBASS lay-out in aircraft cockpit

The ONBASS core unit is connected to the aircraft sensors - basically the Air Data Computer, the Altitude Encoder, the GPS unit and the Slave Gyro - from where it receives data through the Flight Data Interface (FDI). This data is monitored over a period of time and assessed against thresholds and/or predefined patterns etc. by the Flight Data Processing Unit (FDPU), as discussed above, for any hazardous values or trends.

In addition, the FDPU generates a series of instructions/guidance messages for the pilot in case a hazardous situation is diagnosed. These messages are then relayed to the ONBASS HMI unit (basically a PDA or Laptop) where they are displayed on screen for the benefit of the pilot.

The data received from the aircraft sensors is further also stored in the ONBASS core unit's Flight Data Memory (FDM) for post-flight or even long-term trends analysis.

In terms of the system's design & operational aspects, the solution developed by the ONBASS Consortium is illustrated in the following figure, with some brief explanation of the function of each part of the system. As previously discussed, every aircraft can be broken-down into a series of comprising 'elements' (e.g. the landing gear, the engine(s), avionic systems, etc). These all produce a series of data 'packages' (i.e. d1 to dn). Based on these and some 'Flight Mode Rules and Criteria' which the ONBASS partners have developed (which vary from aircraft to aircraft depending on its particular performance or other characteristics) it is possible to define the flight mode which the aircraft is currently in. This data ('packages', flight mode and the associated time stamp) are all then registered in the Flight Data Memory (FDM). From this vast 'repository' of data, using the 'Element Predicates per Flight Mode', the 'Fault Dependency Matrix of Elements' and the 'Expert Rules for Flight Data Management', it is possible to determine the aircraft's fault/risk status. Then, basing on the 'Recovery Methods and Safety Rules' defined, an appropriate response and recovery strategy can be decided upon by the system and the associated 'Advice Profile' (for both faults and safety) is communicated to the pilot (using the 'Language and Symbol Library') through the system's HMI. To close this safety/control loop, the pilot (or ground crew during maintenance) may then provide appropriate safety feedback/control inputs, which will improve on the safety levels of the aircraft in its current state.

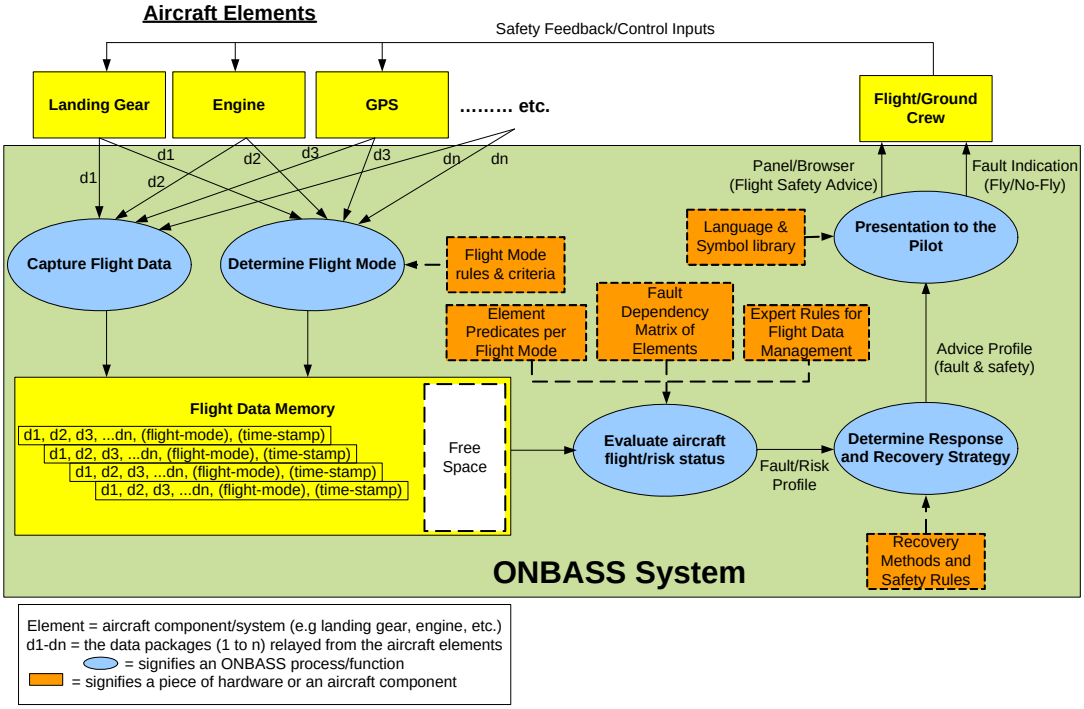


Figure 10: ONBASS System Design & Operational Aspects

The ONBASS consortium has further developed two HW prototypes of the system which complement the software development activities. The first prototype, the famous 'orange' box was developed so as to be used as the platform for the initial development/debugging activities until the final hardware with Fault Tolerant (FT) aspects was completed. This FT HW has been designed for high availability and is

referred to by the consortium as the 'black box'. Both of these prototypes can be seen in the following figure:

Figure 11: ONBASS Hardware Prototypes

The system developed as part of the ONBASS project is customised for the Piper Cherokee Lance (PA-32R-300), a typical General Aviation aircraft. The system can however be relatively easily configured/tailored to the specificities of any other GA aircraft through the 'customisation' of an XML file which holds the relevant values.

The reason why the ONBASS system was customised with respect to the Piper Cherokee Lance was that this is the aircraft with which the in-flight testing campaigns are to be carried-out as part of the project. Apart from these limited flight trials, a series of simulated trials/scenarios are to be used to verify the performance of the system especially in cases where the integrity of the aircraft and its passengers cannot be compromised for the sake of verification/validation of the ONBASS system. The ONBASS demonstrator set-up to be used during these simulated campaigns is illustrated in the following figure:

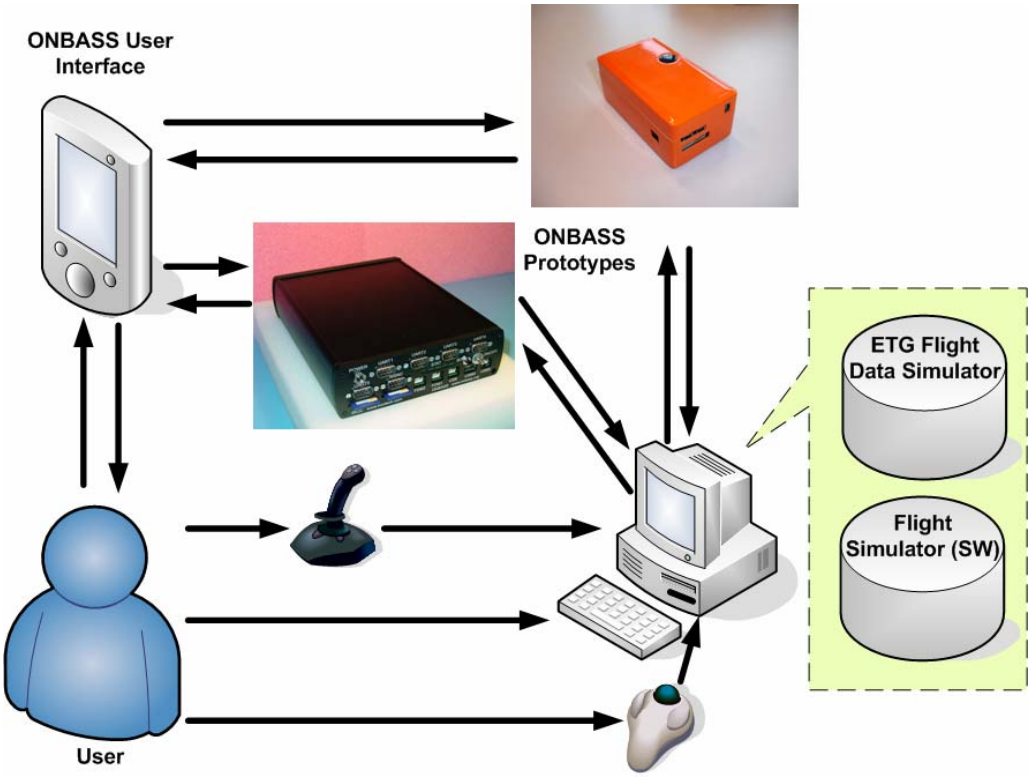


Figure 12: ONBASS Demonstrator Set-up

As part of this set-up, the following elements have been employed:

- Microsoft's Flight Simulator (FS2004) as the data source. A model of a Piper Cherokee Lance is used during the simulated trials.
- The Flight Data Simulator, a tool developed by Euro Telematik (ETG) is used to collect the flight data from Microsoft's Flight Simulator and convert it to a specific standard format that the ONBASS box expects and that typically the real aircraft supplies. This data is then sent periodically, over two serial connections, to the ONBASS box.
- The ONBASS box acquires, parses and processes the data according to a configuration set stored in a XML file on the box (as previously mentioned). The data processing carried-out by the box at this time, has as follows:

    1. The current flight mode (taxi out, cruise, descent, etc.) is determined.
    2. The current flight state is analysed, based on the flight mode previously deduced. The analysis criteria are specified in the XML file previously mentioned (which is either located on the ONBASS box or automatically downloaded via the serial host connection).
    3. If the evaluated flight state is rated as hazardous, a flight advice profile is generated, of which a visualised version is illustrated for the benefit of the pilot on the ONBASS HMI, i.e. a PDA or a notebook.
    4. All fight data collected is time-stamped in a Flight Data Recording stored in solid state (FLASH) memory for post flight analysis and possible longer term archival storage.

5. Technical Results

On the technical side, the ONBASS project has already produced a number of innovative outputs with more to follow. As part of the ONBASS project the consortium has developed:

- The 'Principle of Active Safety System (PASS) algorithm' which basically is a monitoring loop which runs continuously during aircraft operations and completes in a certain order a series of safety enhancement related tasks.
- The 'ONBASS system concept' which in practical terms is the implementation of the PASS algorithm. This is to run as a loop involving the determination of the aircraft current flight mode ('ONBASS Flight Mode Detection algorithm'), the comparison of current and historic (stored previously) data received from the aircraft's sensors against a series of thresholds or predefined patterns ('ONBASS Flight Data Predicates') in order to diagnose/prognose any faults/hazards with respect to aircraft/flight safety, the accessing of the built-in knowledge of the association between the various aircraft (performance) parameters ('ONBASS Dependency Matrix') and then finally the matching of the diagnosed faults/hazards with the most appropriate line of recovery actions ('ONBASS Recovery Matrix').
- The 'ONBASS system architecture' which is the translation into physical terms of the above theoretical models and involves the following innovative elements: the 'ONBASS processor', the 'Fault-tolerant ONBASS RAM', the 'Fault-tolerant ONBASS Flight Data Memory' and the 'Resilient ONBASS Software core'. All of these once integrated constitute the 'ONBASS prototype system'.
- The 'ONBASS Software Architecture Definition' which details the software components of the system, their functions, features, roles and interaction as part of the system's operation.
- The 'ONBASS Hardware Structure Definition' which details the hardware elements of the system, their functions, features, roles and interaction as part of the system's operation.
- The 'ONBASS Software Components' including the Oberon SA Language, the Development Environment, the Compiler, the Minos real time system, the Boot-linker, the Data Acquisition Modules, the Network (TCP/IP) components, the embedded web-server and the User Interface.
- The 'ONBASS Non-FT Hardware Prototype' which was used so as to verify the software components' correct function and integration.
- The 'ONBASS FT Hardware Prototype' which possesses a number of Fault-Tolerant (FT) features, such as hardware failure and software error protection characteristics.
- The 'ONBASS Flight Mode Detector' which initially was devised for the 3 basic flight modes (i.e. take-off, cruise and landing) and later improved to include 7 flight modes (taxi-out, take-off, climb, cruise, descent, landing and taxi-in).

6. Conclusions

The results of the ONBASS project, as demonstrated, could be used to minimise the severity while also identify the 'root' causes of any system faults or flight hazards on-board a GA aircraft, thus aiding in the dramatic reduction of aircraft accidents. Additionally, the project results could also be used to optimise the maintenance schedule of such aircraft and to maximise the potential useful/service life of aircraft parts or the aircraft as a whole, as aircraft elements/parts would only be replaced when the data 'trends' indicate that such an element/part is nearing failure.

Although at present these results have been mostly tailored to the features and particularities associated with General Aviation aircraft (and specifically the Piper Cherokee Lance as aforementioned), the underlying theory and basic principles are common and applicable to all types of aviation, i.e. Military, Commercial, Helicopters.

The ONBASS consortium has discovered during the course of the project, that although the GA field may have been the most appropriate for demonstrating the relevant basic principles, it became evident that there is insufficient relevant flight data available in an electronic format. This hindered the development of representative algorithms for a large set of detectable faults or flight hazard situations. In general, it was found that the limited amount and quality of the electronically available flight data poses a limitation to the extent of useful tasks that the system could carry-out in this case; even so much has been achieved. As the availability, relevance and quality grows e.g. moving towards Commercial Aviation or even further to Military Aviation, more tasks could be carried-out. On the other hand, obviously the complexity and size of underlying algorithms will also grow in such a case. The experience from the Transport sector already indicates that a rapid increase in the sophistication of data monitoring and processing at low cost is possible.

The project consortium is currently also further extending the theoretical foundation and more specifically the aircraft/elements/sub-elements/components models of the ONBASS system with a view to make these fully generic and as smart as possible so that they basically continuously 'learn' throughout their operational life.

The long term target of the ONBASS Team is to develop a system that will not only decide on the optimum course of action in the case of a system fault or a flight hazard and provide analogous guidance/instructions to the pilot allowing him to 'close' the loop, it will further via the Flight Management System provide appropriate corrective actions/inputs itself. This way the ONBASS system will become a fully active safety system for all types of aviation.

For all the above, the ONBASS Team is dedicated to further pursue through research the extension and validation of the results of this project to the other fields of aviation (i.e. Military, Commercial, Helicopters) while also at a larger scale and an increased level of complexity and data availability.

7. References

[1] ONBASS Project – 'Description of Work', Version 1.3, 09/01/2007

[2] ONBASS Project – D1.2 - 'PASS Functional & Reliability Models', Version 1.1, 2005

[3] ONBASS Project – D2.1 - 'System Concept and Structure Definition', Version 1.2, 19/7/2005

[4] ONBASS Project – D2.2 - 'System and Application Specification', Version 1.0, 19/12/2005

[5] ONBASS Project – D5.4 - 'System Evaluation Report', Version 0.2, 2007

[6] ONBASS Project - ONBASS Newsletters 1-4, http://www.onbass.org/ , 2005-2007

[7] Schagaev I., CoDySa – 'Concept of Dynamic Safety', Proc International System Safety Society, Seattle, 1998

[8] Schagaev I., Overtoon L., 'Active Safety System for General Aviation', Proc.17th International System Safety Conf., Orlando, Florida, 1999

[9] Schagaev I., 'Concept of Active System Safety', Proc.15th IFAC Symp. on Automatic Control in Aerospace, Bologna/Forli, Italy, 2001

[10] Schagaev I., Kirk B., Bukov V., 'Applying the Principle of Active Safety to Aviation', 2nd European Conference for Aerospace Sciences (EUCASS), July 2007

[11] V.N. Bukov, V.A. Chernyshov, B. Kirk, I. Schagaev, 'Principle of active system safety for aviation: Challenges, Supportive Theory, Implementation, Application and Future'.

International Conference “NEW CHALLENGES IN AERONAUTICS” ASTEC’07, August 19–23, Moscow, 2007.