Project no.: 516045
Project acronym: ONBASS
Project title: ON-Board Active Safety System

Instrument: Specific Targeted Research Project (STREP)
Thematic Priority: Aeronautics and Space

Publishable Final Activity Report

Due date of deliverable: 15 February 2008
Actual submission date: 18 April 2008

Start date of project:	**1 January 2005**
Duration:	**36 months**
Organisation name of lead contractor for this deliverable:	**Euro Telematik AG**

Revision 1.0

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Distribution List

Member Type	No.	Name	POC	Distributed[1]
Web		Internet	www.onbass.org	
Contractor	1	ETG	Thomas Wittig	X
	2	Londonmet	Igor Schagaev	X
	3	IrocTech	Marc Derbey	X
	4	Robinsons	Brian Kirk	X
	5	ETHZ	Juerg Gutknecht	X
	6	SPIRIT	Angus Kintis	**X**
Sub-Contractor		N.N		
		N.N.		
Customer		EU	Jean-Luc Marchand	X

[1] Please insert an **X or** ✔, when the PoC of a company receives this document. Do not use the date of issue!

Change Control List (Change Log)

Date	Issue	Changed Items/Chapters	Comment
2008-01-31	0.1	Initial Draft	
2008-02-28	0.2	Final draft for review	
2008-03-03	0.3	Minor comments by SPIRIT	
2008-04-18	1.0	Release	

Table of Contents

1 Project Execution

1.1 Overall Project Objectives

Due to growth of complexity and cost of aviation operations, caused by the increase in the number of aircraft and air traffic, it has become clear that in the not too distant future it will be impossible to provide and maintain appropriate levels of flight safety with the current safety systems and aviation infrastructure. Complex technical and economic issues just compound the problem and so there is a pressing social demand to:

1. Make effective use of the available experience and flight data to achieve higher safety levels in aviation;
2. Develop rigorously correct hierarchic flight safety systems, which can react to undesirable situations in real-time, thus avoiding unfortunate events and preventing some accidents, rather than just analysing them after an incident has occurred.

To match this demand the scope of this project is the following:

1. Further theoretical and conceptual development of the active safety principles for aviation and formation of theoretical model(s) to analyse the limits of this principle's applicability;
2. Research and development of basic fault tolerant hardware elements for the on-board part of the active safety system for aviation;
3. Concepts, design and development of a resilient system software core for the active safety system for aviation;

This scope can be achieved by work in the following closely related areas:

- Researching of operational model and development of a theoretical model of flight risk;
- Development of rigorous system requirements for the realisation of PASS;
- Research, conceptual design and development of system software;
- Research, conceptual design and prototype development for on-board embedded hardware;

Given the limited budget of a STREP and the fact that safety data and details of safety system structures are handled somewhat defensively by aircraft manufacturers and are very unlikely to be made available to the ONBASS consortium, **the work will primarily focus on the creation of a new class of product for a new market – active safety systems for General Aviation.** The advent of the availability of private jet aircraft at sub $1 million prices by 2006 brings further urgency to the situation. Currently neither regulations nor flight safety systems exist in General Aviation and so it seems to be a good choice for the initial verification of the ONBASS concept and prototype.

All activities will be accompanied by detailed supportive schemes of dissemination and further industrial European implementation of the project results.

In summary, the scientific and technological objectives of ONBASS are:

- To define the theoretical limits for safety improvements by means of proper use of available flight data, i.e. how and how much information about a flight can be used to provide improved safety;
- To suggest schemes and the means by which aviation safety systems should be organised so as to enable effective flight information processing;
- To analyse the complexity of real-time flight data and how to make it simpler;

- To establish the required performance and reliability levels for the active safety system and its main elements (on-board, on-ground, National/International/European level);
- To investigate the role of the main "agents" responsible for risk and theoretical ways to tolerate/address them;
- To research the software aspects associated with robust and reliable safety-critical on-board software (specification languages, formal methods and tools);
- To conduct analysis and modelling of essential features and elements;
- To design and develop the system software structures for active safety systems;
- To analyse, model, design and develop the hardware for active safety systems;
- To investigate, analyse and define the economics and business aspects associated with the PASS approach in the long-term.

1.2 Project Consortium

The ONBASS partners span much of the very diverse geographical and cultural background of the European Continent. The ONBASS partners have been brought together to complement one another in terms of technical expertise and scientific excellence. Together they are considered to amount to the 'critical mass' required to ensure that both the technological and scientific aspects associated with the ONBASS project are tackled in an efficient and effective manner. ONBASS requires broad skills from all project members – collaborators in science, software and hardware theories, software and hardware development, technology development and application, industrial and practical experience in IT application for large-scale special purpose systems.

The consortium, therefore, is composed of a balanced mixed of large and small companies as well as universities. The work share is spread over four member states of the European Union (Germany, United Kingdom, France and Greece) and one associated state (Switzerland) for the 6th Framework Programme. The involvement of 3 SMEs (1 from France, 1 from Greece and 1 from the United Kingdom) ensures that the ONBASS project will reinforce the European policy for the involvement of SMEs in such EU co-funded research. These SMEs have been selected to participate in the ONBASS project for their highly specialised expertise in the field. The table below further summarises the consortium composition.

No.	Partner	Short Name	Business Domain	Contribution
1	Euro Telematik AG (Germany)	ETG	Expert in R&D related to flight management, JAR-certified avionics developer and manufacturer	Project Co-ordinator, responsible for overall system design, specification and verification
2	London Metropolitan University (United Kingdom)	Londonmet	University with high scientific and practical background in aerospace and IT	Theory and models of active system safety, structure and main part of active safety system, theory fault tolerance
3	iRoC Technologies (France)	iRoC	SME, specialised in the design and building of high integrity hardware systems	Responsible for hardware system design and specification, verification and validation
4	Robinson Associates (United Kingdom)	Robinsons	SME, specialised in the design and building of high integrity software systems	Contribution to system design, specification, simulation, verification and validation
5	ETH Zürich (Switzerland)	ETHZ	World Leading Institute of Technology	Theory of system software for safety critical systems, modelling of reliability and safety
6	SPIRIT (Greece)	SPIRIT	SME, specialised in aerospace related system engineering	Contribution to system design and specification, verification and validation

2 Project Activities

2.1 Theory and Operational Model

2.1.1 Overview

The objective of this work was the development of theoretical and operational models of aircraft exploitation in terms of a formation of a safety profile and monitoring. After a systematic survey of the application domain and the processing of existing statistical data in the application domain, the profile of flight risk for Commercial and General Aviation was developed. Having this data in combination with the analysis of existing systems available, a conclusion was made about features of operational models that will enable an operational risk analysis in real-time of flight. From the operational risk analysis model a reliability model of flight was derived, aiming at the possibility of real-time prognosis of flight risk. Finally, the programming of the reliability model was done and a simulation of its operation in real-time data processing was given.

2.1.2 Application Domain Definition

This section summarises the results of the ONBASS Application Domain Definition.

Survey of Application Domain

The survey started with a short introduction of the terminology for the classification of aviation as well as active safety. The classification of aircraft according to its mission, the type of aircraft, the consumer properties and the state of development were covered. While it has to be noted that all classification approaches are subjective, still the outlined classifications are suited for the purpose of considering the impact of active safety with a target on General Aviation.

This work was followed by a market overview about aviation, including military, commercial, general aviation and helicopters. This overview shows clearly that the numbers for aircraft in service world-wide will further grow leading to increasing challenges on the future Air Traffic Management and related safety aspects. The overview further visualises that General Aviation aircraft have a significantly high share in the overall aircraft market. Various sources are further analysed with regard to GA market and statistical data. It was pointed out that around 300.000 GA aircraft today exist world-wide. The US, Canada, Germany and Australia are highlighted as the main markets for this kind of aviation. In this context, models and manufacturers of GA aircraft have been analysed as well showing how widespread the categories of use of GA aircraft are. The overall goal of ONBASS to realise an onboard active safety system that is targeted at an improvement of flight safety in the General Aviation domain is, therefore, perfectly in line with the respective market characteristics. The review concluded with an analysis of civil aircraft shipments during the last 15 years.

Trends and Strategies in Air Traffic Management

Given the fact that the number of aircraft is still expected to grow, immediate attention is required world-wide. In the recent past, and certainly for a number of years to come, a great deal of effort has been or will be spent on the development of CNS (Communications, Navigation and Surveillance) techniques in order to generate enhanced ATM (Air Traffic Management). For that reason, the current trends and strategies in Air Traffic Management were summarised. The aim of all these activities is the provision of enhanced ATM services and operations between now and the year 2020 by the application of a series of incremental ATM operational improvements. The generalisation of best practices and their systematic use will generate continuing improvement in terms of safety, security, capacity and the environment. This evolving ATM environment was taken into account during the ONBASS system specification & design.

Risk of Flight

Emphasis was put to provide an overview of the typical flight phases and the possible risks during these flight phases. In this context, the main risk factors were correlated to the respective flight phase. Several aviation accidents were summarised and analysed. A short overview was further given on the involvement of the insurances in today's aviation.

Risk and safety in GA was further detailed. This started with an overview of safety management in GA, highlighting the problem of establishing a clear unified framework for GA operations internationally. Several accident statistics were presented, showing that accident rates and number of fatalities associated with GA are still considerable. It should be noted that 67.5% of all GA accidents were caused by controlled flight into terrain, loss of control in VMS, low flying/aerobatics and loss of control in VMC. Based on the accident analysis, GA related flight risks have been analysed. Most causes and factors attributed to GA accidents were personnel related. The majority of causal factors had a Human Factors element whether in terms of pilot or maintenance actions. An interesting observation is that fatal accidents were more related to the speed of the aircraft rather than the phase of flight itself.

Analysis of Safety Systems in other Domains

To broaden the background for the subsequent ONBASS system definition, an analysis of safety systems in other domains where safety is considered a critical aspect was made. Safety systems in industrial systems / machinery, automotive industry, rail and space are reviewed and relative standards in all four domains were examined.

Avionics Architectures and System Components

This activity started with an overview of avionics trends in GA. It concentrated on new GA ex-factory aircraft models referring to the considerable and far-reaching changes which General Aviation is currently undergoing, and which are set to initiate a new and ongoing demand, in the mid-to long term, the like of which has not been seen in the field for many years. Regarding these avionics developments it can be stated that several sensors which are already onboard these aircraft can be used for the purpose of the ONBASS project.

Subsequent work was focused on existing on-board safety systems in GA. It included an overview of existing technologies and systems related to safety like TCAS, GPWS, ELT or Flight Data Recorders, all of which are passive safety systems. Current safety systems like TCAS, GPWS or Flight Data Recorders are in the price range of around 30,000 USD each, which is too expensive for the majority of General Aviation users.

For this reason the respective avionics components applicable for this project were described in more detail, covering navigation equipment, fuel/air data computers, engine trend monitor, altitude encoder and slave gyro. The input and output parameter values from these systems were listed.

For the purpose of ONBASS the following avionics systems are of relevance:

- GPS/Navigation unit: almost all GA aircraft are equipped with GPS. Via a standard NMEA interface the respective GPS data can be accessed (Latitude, Longitude, GPS altitude, GPS time, …)
- Air Data Computer: an increasing amount of GA aircraft are now equipped with an Air Data Computer that delivers a number of aircraft parameters.
- Altitude encoder: all GA aircraft are equipped with an altitude encoder delivering barometric altitude data via a serial interface or via a gray code interface
- Slave Gyro: around 40% of GA aircraft are equipped with a Slave Gyro system delivering the heading value.

Based on this analysis, the following figure shows a potential architecture for GA that could build a reference for the ONBASS project and served as an input into the ONBASS system design. Based on the required ONBASS data, this architecture might by applied in respective subparts.

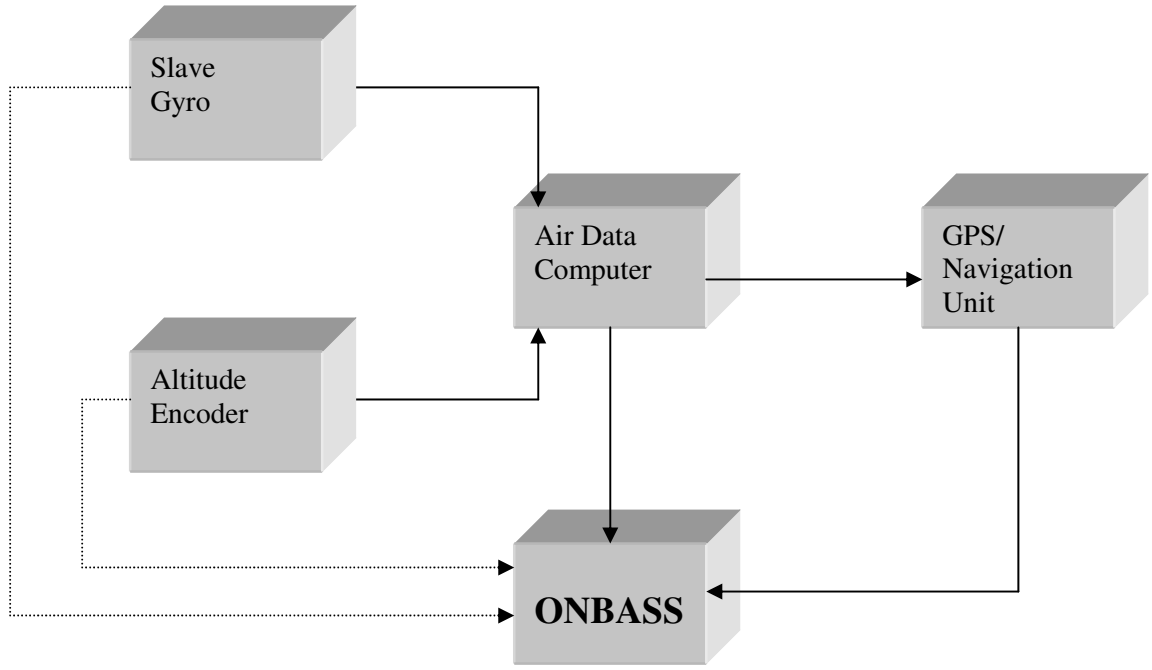


Figure 2.1: Potential GA architecture for the integration of ONBASS

Aviation Safety and Data Recorders

Taking into account the analysis of onboard systems as described before, this work now concentrated on data recorders in particular.

The main actors in aviation safety were discussed and the main technological and structural approaches for data recorders were described, starting with an overview of data recorders in the overall transport domain. The work then focused on aviation and examples were presented for current systems in General Aviation and Military Aviation, also taking into account the post-flight data analysis.

The typical data flow of safety related data on-board an aircraft as well as the flight data exploitation cycle was outlined. Finally, flight parameters which may be recorded onboard in different types of GA aircraft were presented.

As far as it could be seen, there is no focus across the whole of aviation on safety systems or devices which would be designed to provide safety 'actively' and consistently.

The aim of the ONBASS project is to address the issues outlined above in the context of European General Aviation by developing the underlying theory and technology specifics associated with *Active Safety*.

2.1.3 Reliability Model

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 Figure 2.2 below:

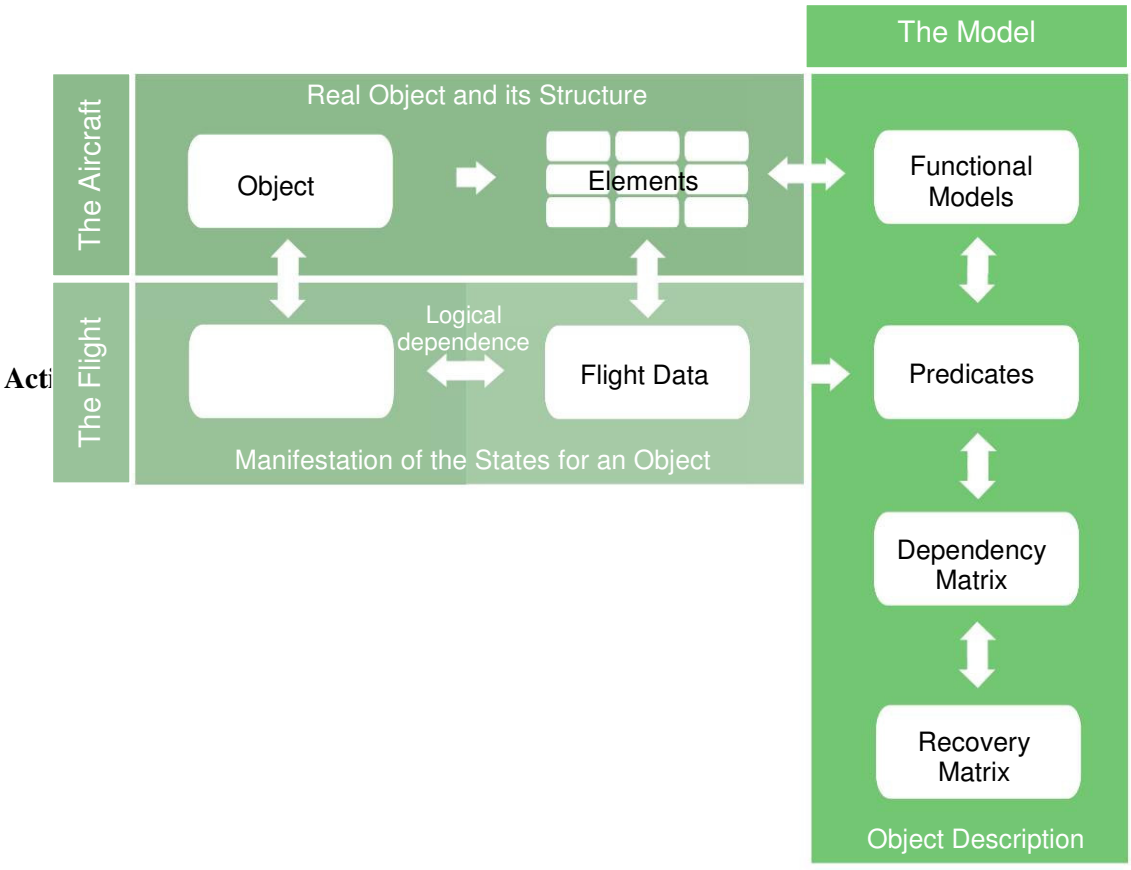


Figure 2.2: Structure of the Model of Active System Safety for Aviation

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 2.2 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 2.3 below:

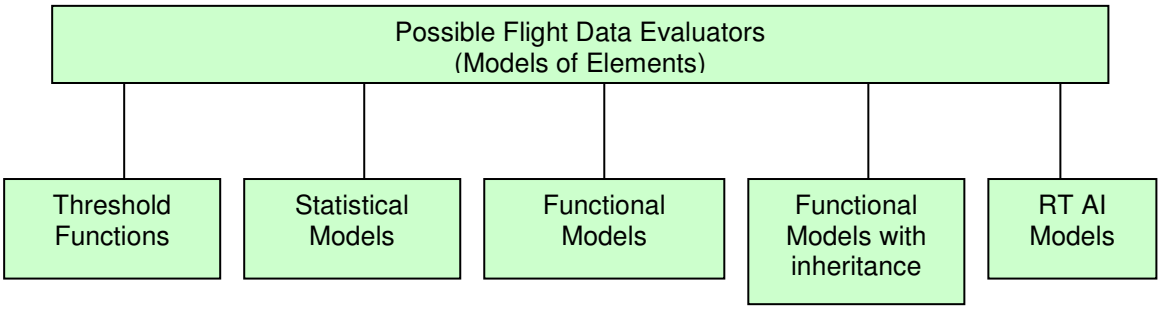


Figure 2.3: Possible element modelling techniques

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 2.4).

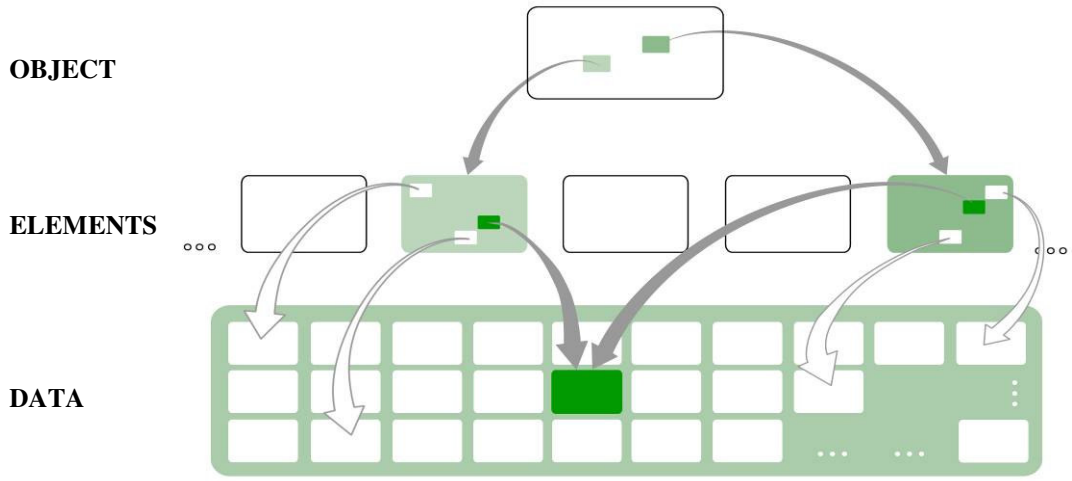


Figure 2.4: Dependencies of data and elements

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 2.5 and 2.6 that follow:

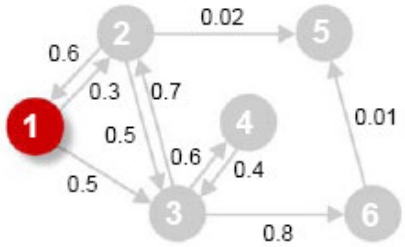


Figure 2.5: Probabilistic Graph Dependency Matrix representation

	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 2.6: Probabilistic Dependency Matrix representation

In the graph of figure 2.6, 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.

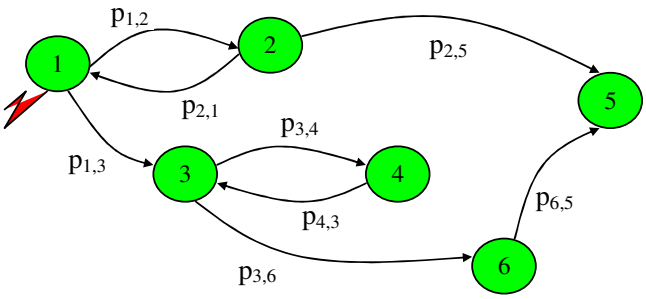
The alternative ways to react to a hazardous condition that the object (aircraft) is subjected to, arising from the specific condition of all the object's elements, are defined in the Recovery Matrix. The Recovery Matrix maps one-for-one to the Dependency Matrix, (i.e. if the Dependency Matrix has dimensions *n*x*n*, the Recovery matrix will also be a *n*x*n* 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 figure 2.7 and include the Dependency matrix and a directed probabilities graph.

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 figure 2.8 defines how the Recovery Matrix is populated initially, how it is used, as well as how it is thereafter updated.

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.



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

Figure 2.7: An example of the 'reverse tracing' process

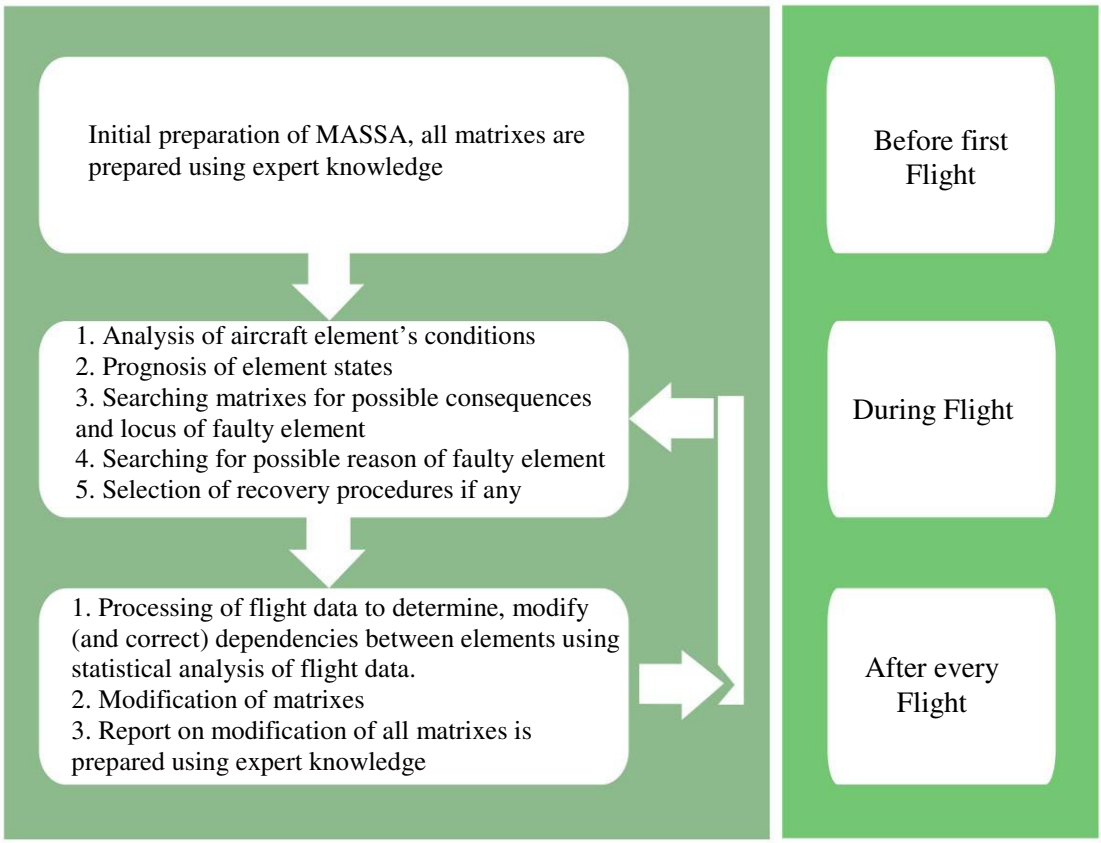


Figure 2.8: Creating, using and updating MASSA

2.2 System Requirements Analysis

2.2.1 Objectives

The objectives of this work were to:

1. Clearly and concisely define the overall requirements of the system both from the external point of view of its users and also in terms of its internal function.
2. Ensure that the safety context and safety requirements are clearly defined.
3. Understand which Certification and Qualification standards are relevant and to what extent.
4. Define what the system must do, and must not do, for each distinct kind of user and what interaction there is with the user, and how it will be represented.
5. Define the interfaces both externally to the devices operational context, internally within the device between subsystems, and within subsystems between components.
6. Define the hardware architecture to provide the reliability and fault tolerance required.
7. Define the system software (runtime) required to support the hardware and to present a resilient set of services to the application software. Define the application software required to meet the needs of each kind of user.
8. Ensure that the design is traceable and verifiable by audit, and that the prototype's function is testable in practical terms.

2.2.2 System Architecture and Design

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:

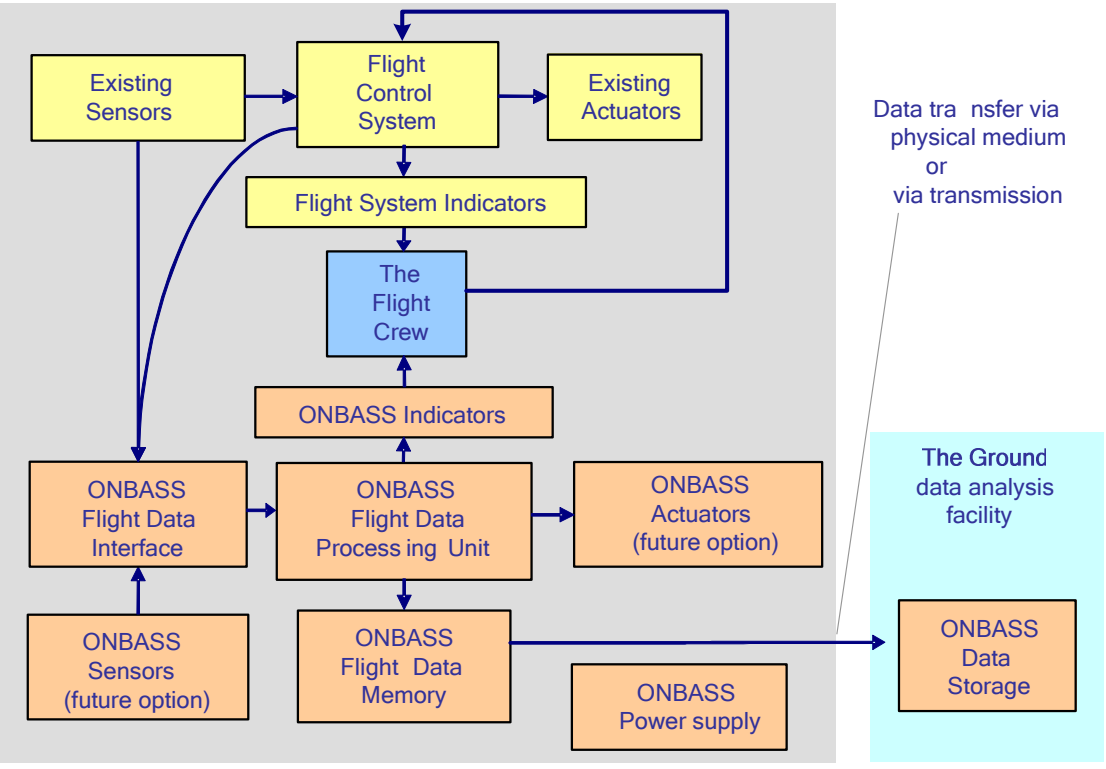


Figure 2.9: ONBASS System Architecture

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 2.10: 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.

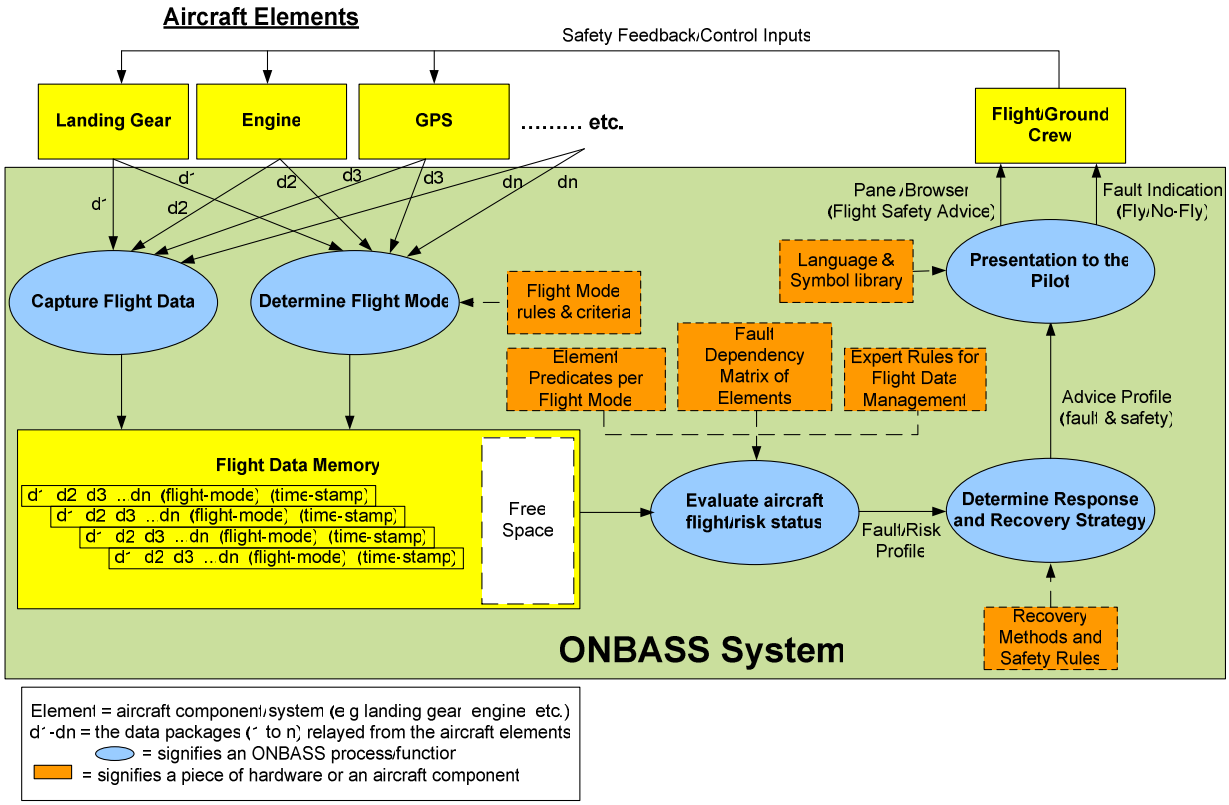


Figure 2.11: ONBASS System Design & Operational Aspects

2.2.3 Certification Aspects

Requirements and guidelines for certification of avionics equipment under the authority of the European Aviation Safety Administration (EASA) are specified in documents published by EASA as "Decision" documents.

For TSO equipment, the applicable document is "Decision No. 2003/10/RM" that contains the EASA TSOs (called ETSO). These TSOs (Technical Standard Orders) use the same numbering as their US counterparts published by the FAA. In some cases, however, the ETSO will vary from the corresponding FAA TSO. And there are some ETSOs and some FAA TSOs that do not (yet) have a counterpart from the other agency.

(E)TSOs are the top-level specification of minimum requirements on equipment for certification. In most cases, they do not contain a lot of detailed requirements but refer to industry standards published by various organisations, including the RTCA group, EUROCAE, or ARINC.

In order to assess the necessary requirements for certification of a system such as ONBASS, these referred standards must thus be examined.

Regarding hardware qualification, the standard that is referred to most often is RTCA DO-160D (or its European counterpart, EUROCAE ED-14E). This standard specifies a variety of tests (such as temperature, altitude, and EMC tests) and for each test a set of test categories. The selection of the applicable tests and test categories may either be specified directly in the (E)TSO or in one of the referred specifications, or the manufacturer may select these categories. In the later case, DO-160D/ED-14D provide guidelines on that selection, depending on the expected location of the equipment inside the aircraft.

Software Development is most often required to follow the guidelines of RTCA DO-178B (or EUROCAE ED-94B). This standard defines five levels of design assurance, ranging from level E (for equipment without any software-related safety concerns) to level A (for highly safety-critical equipment such as autopilots). The selection of the applicable level usually lies within the responsibility of the system developer as a result of a safety analysis. In some cases, however, the (E)TSO or the referred standards may specify a minimum design assurance level that must be applied.

Concerning the Minimum Operational Performance Requirements (MOPS) for Data Flight Recorder the focus lies on the document ED-112. This document supersedes the MOPS for Flight Data Recorders (ED-55 or TSO-C124) and Cockpit Voice Recorders (ED-56A). The MOPS specified in this document are based on the Standards and Recommended Practices (SARPS) published by ICAO in Annex 6 of the Chicago Convention. Account has been taken of regulations issued by various certification authorities together with the advice and recommendations given by accident investigation specialists. The document represents the minimum standards required for the purposes of accident investigation.

Given the limitations of a R&D program, in the frame of the ONBASS project only a subset of the respective requirements can be implemented. This, in particular holds true for the development and test of software and hardware.

Although the certification guidelines request a SW development process according to DO-178B Level C, ONBASS will restrict the work to a Level D like development process.

With regard to the hardware, the tests according to DO-160D as described before can not be performed under the given financial and time constraints.

The main restriction with regard to the overall system functionality according to ED-112 exists in the recording of the requested data. Again, only a subset will be implemented, taking into account the respective application of the system in General Aviation and the nature of this R&D project.

The recorded data will in maximum include the following parameters:

- IAS (indicated airspeed)
- P.ALT (pressure altitude)
- OAT (outside air temperature)
- HEADING
- IVS (indicated vertical speed)
- TAS (true airspeed)
- MACH
- WIND SPEED
- WIND DIRECTION
- FUEL FLOW
- GPS Position Data (latitude, longitude, ground speed)

2.2.4 System's Applications

An overview of the ONBASS system's application is shown in Figure 2.12.

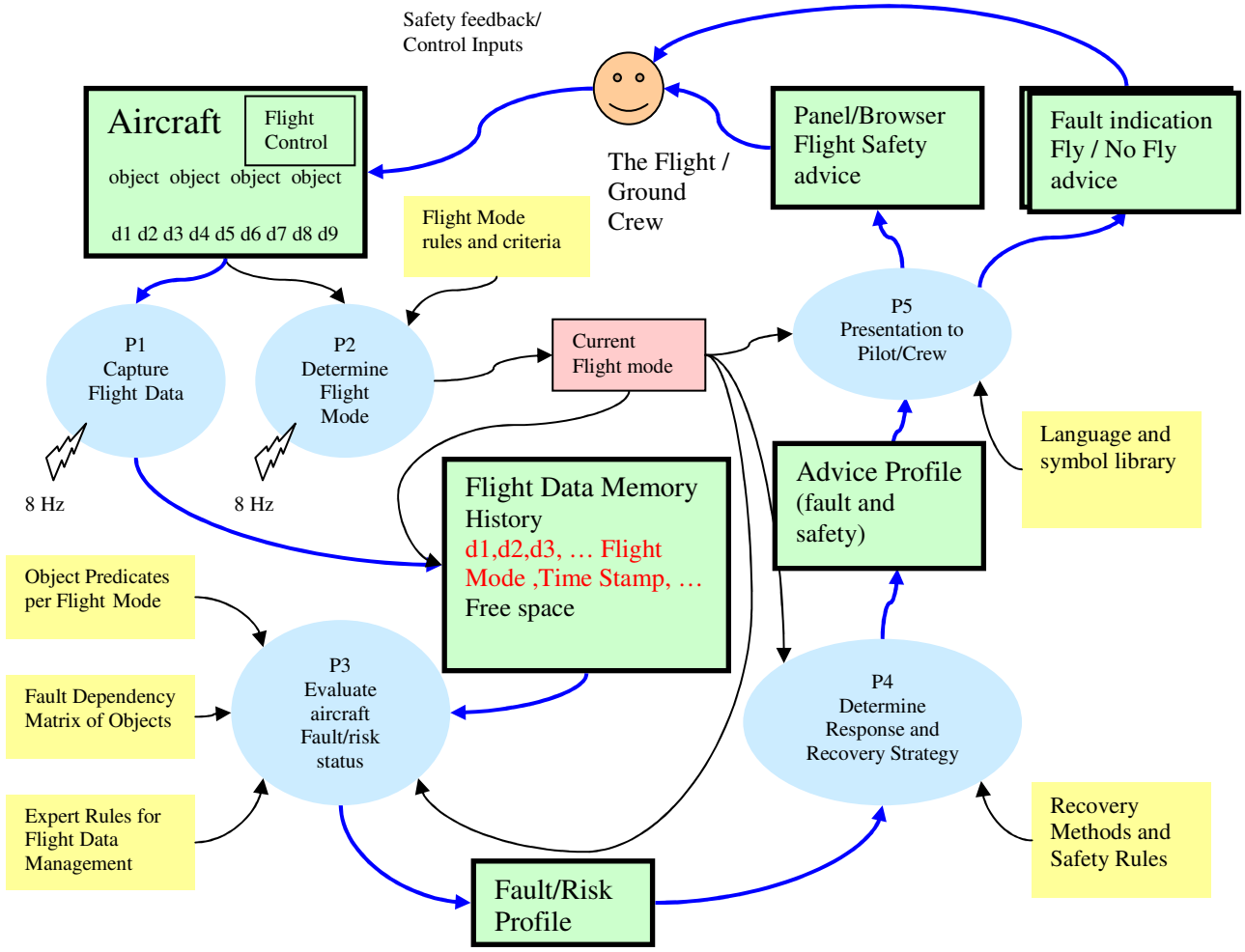


Figure 2.12: The ONBASS Active Safety

The conventions used in Figure 2.12 are as follows: relatively static data is shown in rectangles, data that is flowing are vertices and information processing is represented by circles; the latter are numbered to ease reference for the description below.

Fault Management Control Loop

The ONBASS system is basically a fault management control loop at the level of the aircraft itself. Its purpose is to improve the safety of flight either by advising against starting a new flight or providing salient advice in flight in the event of a fault developing which might compromise safety. The information processing involved in the loop is shown in Figure 2.12, and represents an abstract view of the top level implementation of the PASS/reliability model within the aircraft during flight, in real time. Note also that the system is active, with the same analysis, before takeoff and so may offer an 'unsafe to fly' indication to the ground and/or flight crew. In this case it would be unsafe to attempt a flight at all.

As per the 'Reliability Model Description' the aircraft is considered to be composed of a set of objects (e.g. engines, landing gear, brakes , flaps, GPS …) which contain data sources (e.g. sensors) producing a time series of data values during operation of the aircraft.

Flight Data Memory

Data is captured from the aircraft's sensors and avionics, via the Flight Data Interface (FDI) on a periodic basis by P1 and stored in the Flight Data Memory (FDM). The frequency of the data capture will initially be 8Hz, this being an established standard for flight recorders. However, when real data is available it will be analysed to determine the frequency required in order to be able to extract the required information from the data. In this way it will be possible to optimise the amount of relevant data that can be stored in the limited Flight Data Memory. The FDM has 3 zones, the current data record being written, a history of previous data records and free memory space available for writing future data records. It is anticipated that the storage will be structured as a circular buffer of data records so that the FDM is never full; each new record overwrites the currently oldest record. Note that each data record includes a time stamp and an indication of the current Flight Mode. The ED-112 specification of data stored in standard aircraft data recorders has been mentioned, ONBASS however will use only a small subset of these parameters.

Flight Mode

The modes of flight have already been mentioned. There are a variety of conventions regarding its granularity in different countries e.g. in the USA, UK and Germany. For the purposes of ONBASS 7 flight modes are considered; these are: Taxi, TakeoffGround, TakeoffAirborne, Climb, Cruise, Descent, Landing and Taxi. These modes have significantly different risk profiles and the appropriate responses to cope with faults may differ depending on the flight mode. The flight mode is determined continuously in real time by P2 based on some rules and criteria that are specific to the aircraft's flight characteristics (e.g. takeoff speed, landing speed, vertical speed, engine revs, etc.). The rules and criteria for determining it must be configurable. A suitable periodic time interval for determining the flight mode is currently thought to be 1Hz, however, the conventional rate is 8 Hz for conformance with ED-112.

Real Time Fault Diagnosis and Prognosis

The evaluation of the current (diagnosis) and future (prognosis) faults in the aircraft is performed by P3; this takes data from the FDM and:

1. Evaluates the fault status of each of the aircraft's objects based on a set of predicates, then,
2. For each fault it uses the fault dependency matrix to assess consequential contributions to other fault conditions using,
3. A set of expert rules (packaged experience) to guide and characterise the evaluation.

The net result of the evaluation is a fault profile which defines the current reliability diagnosis/prognosis in terms of a set of objects, each with actual or pending faults.

The three data sets, listed above, are each specific to the aircraft's reliability characteristics. Their content is based on an analysis of the reliability model of each kind of aircraft and is downloaded as part of the ONBASS configuration process.

Determination of Recovery/Response Strategy

The response to the fault/risk profile is generated by P4 to determine how best to conserve or improve safety by considering the whole fault/risk profile and:

1. Mitigation of the effects of a set of current faults/risks.
2. Avoiding or preventing escalation of current faults from developing into future errors or failures.
3. Preventing the likelihood of occurrence of pending faults/risks, or mitigating the severity of their effect.
4. Addressing fault/risk combinations by priority (i.e. higher risks first).

This involves evaluating the fault/risk profile from P3 based on so-called recovery methods and a set of rules, the flight mode characterising the overall strategy. The result of this evaluation is an advice profile aimed at conserving or improving the safety of flight i.e. a set of information and recommended actions for the pilot.

There are two kinds of information resulting from this analysis:

1. Fault advice e.g. major fault detected; aircraft not airworthy.
2. Consequential safety advice based on a safety analysis of the consequences of faults e.g. fuel leak in left engine, shut down engine if possible (to avert a fire).

The safety advice referred to above is generated from a safety analysis of the consequences of single and multiple faults/risks. This will be specific for each type of aircraft, hence the need for customisation.

Presentation to the Flight/Ground Crew – closing the loop

The set of fault and safety advice is formatted for presentation by the Human Machine Interface (HMI) to the pilot by P5. The communication has to be clear, concise, precise, relevant, practical, timely and easy to assimilate. The pilot observes the advice and uses his/her judgement and skill to decide how best to adapt the control of the aircraft given the current perception of the flight context and the safety advice from ONBASS. The pilot's reaction to the advice, via the Flight Control System, closes the safety control loop on board the aircraft in real time, depending of course on the response time and appropriateness of reaction of the pilot.

There are 2 kinds of indications envisaged:

1. A simple 'Fly' /'No-Fly' indicator to warn the flight and/or ground crew that the aircraft is not considered airworthy due to a current or impending fault (set) which might render it unsafe.
2. Advice to the Flight crew in the event that a fault is diagnosed, or prognosed, during flight which would impact the safety of the aircraft. It is envisaged that this information relates to safety i.e. the avoidance of harm occurring during the rest of the flight.

A browser will be used for display to provide flexibility in the choice of display panels e.g. PDAs.

The representation may also be configurable for colour rendering and graphic symbol imagery.

2.2.5 System Verification Approach

This section summarises the results of the ONBASS Verification and Validation Plan.

The verification tasks shall ensure that the specification and implementation of the ONBASS prototype is traceable and auditable across its lifecycle, and that verification audits are performed. It is also concerned with ensuring that the prototype is testable from the view point of each kind of user so that it can be validated, and to define a general format for test specifications.

It should be noted here, that the main activities within the ONBASS project do concentrate on a verification of the system and application requirements. Due to the limited resources, validation will be only a minor activity.

Acceptance Testing will ensure that the ONBASS demonstrator complies with the applicable requirements as set forth in the ONBASS System and Application Specification. The output of the software design, coding, and integration processes is verified against the respective software requirements. The detailed test procedures for each requirement have been specified.

Scope of Verification

During a Factory Acceptance Test (FAT), all requirements set forth in the ONBASS System and Application Specification will be verified. While some requirements will be directly verified through appropriate test procedures, other, high-level requirements will be verified indirectly by conducting tests for low level requirements that can be traced back to those high level requirements. The Site Acceptance Test (SAT) will include a subset of the FAT tests and some additional tests (for the aircraft SAT) that can only be tested in the real aircraft environment.

As part of the FAT, those tests that had to be conducted in advance of the FAT (such as software module tests prior to integration and hardware qualification tests conducted at external laboratories) will be examined by reviewing the test documentation for those tests (plans, procedures, records, and reports).

Verification Methods

The purpose of verification is to test if the result of a process (e.g. a document or software code) meets the requirements specified for the item. To fulfil this purpose, there are several methods available:

- Document Review
- Code Inspection
- Software Module Testing
- System Testing
- Hardware Qualification Testing

Once the requirements have been established, the verification team needs to determine which testing method is appropriate for each requirement and then specify a test procedure that will verify that the requirement is fulfilled. To ensure that all requirements have been tested, traceability matrices need to be created that show which requirement is covered by which test procedure and vice versa.

Responsibilities

Verification will be conducted by members of the development team at various stages.

Once the System and Application Specification has been released for review, the verification team needs to check the requirements for completeness, correctness, and consistency.

As soon as this allocated baseline is established, the verification team specifies detailed test procedures for the software that will allow to verify the correctness of the code and its compliance with the System and Application Specification.

The software and hardware will then be verified by testing the integrated system against the requirements specified in the System Specification, following these test procedures. In addition, individual software and hardware modules may be tested prior to integration. These tests will also be documented in the FAT/SAT Procedures.

2.3 Software Development

2.3.1 Objectives

The objectives of this work package were to:

1. Design and implement a new generation of system software for embedded real time systems with a focus on support for hardware fault tolerance, reconfigurability, consistency self-checking and recovery during operation
2. Develop new integral design methodology for developing such systems, improving the state of the art in terms of correctness, robustness and resource efficiency with an emphasis on hardware/software co-design
3. Develop recovery techniques within the real time system to support dynamic reconfiguration of the hardware during and after fault detection/ reconfiguration, whilst preserving state and synchronisation in a rigorous way
4. Apply the aforementioned systems to safety-critical aviation control and demonstrate with the PASS application

2.3.2 Software Architecture

The entire ONBASS software can be grouped into two major parts:

a) The onboard computer runtime and
b) Development tools.

The following figure 2.13 is intended to give a structural overview of the onboard computer runtime. Each component or *module* (represented by a box) implements some specific functional domain and exports an *abstract interface*. Upper level modules may use or *import* (interfaces of) lower level modules.

The runtime software can roughly be partitioned into three parts, the *runtime core*, the *application framework*, and the *applications*. The *runtime core* called *Minos* corresponds to the actual "operating system". It is responsible for the management of resources such as processor, volatile memory, flight data memory and I/O ports. In addition, the runtime core provides file system functionality and low-level recovery procedures, and it initialises the system plus its components.

As depicted by the above diagram, the runtime core is hierarchically structured. Components close to hardware are allocated in lower levels. The lowest level is the *Hardware Abstraction Layer (HAL)*. Its purpose is improving the system's portability by hiding platform-specific details. Other low-level components are the *system boot loader* and the *floating point emulator*.

The *device drivers* are allocated on top of HAL. They use the HAL abstraction to communicate with hardware devices. On the next upper level we find the *I/O system* whose responsibility is standardizing input/ output programming. The *memory management* system manages the system heap. It provides methods to get and return memory blocks. However, the ONBASS runtime core does not implement automatic garbage collection because it is de-facto incompatible with hard real time constraints. Instead, an explicit set of rules will encourage programmers to employ a specific programming discipline with the goal of avoiding garbage.

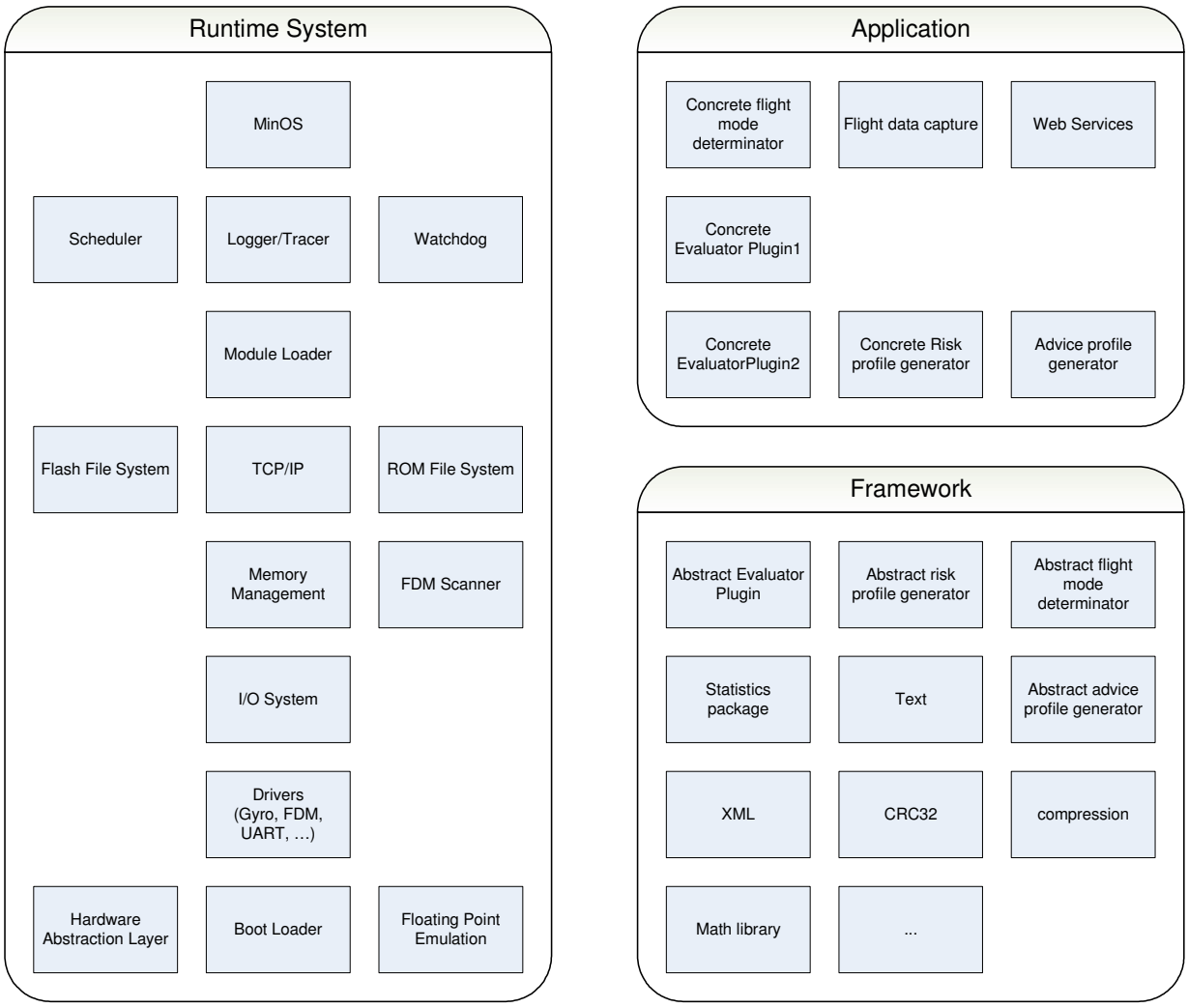


Figure 2.13: Onbass Component Overview

Also required are implementations of several protocols and of other services including, for example a ROM file system, a flash RAM based file system and TCP/IP support. Dynamic loading of software components is supported by a linking *module loader*. The *system scheduler* is responsible for distributing the processor among the different runnable tasks according to the imposed real-time constraints. Important auxiliary tasks are a *liveness checker ("watchdog")* and a *logging and tracing facility* for debugging purposes. The top level component called *Minos* basically provides the user interface functionality.

The *application framework* consists of a) a plug-in mechanism for flight-data analysis algorithms and b) a software library providing generic support for various topics, including *mathematics* and *statistics*, *strings* and *XML document management*. This list of topics will probably have to be reviewed and extended during the course of the implementation phase.

The *application* itself basically consists of an implementation of the PASS algorithm. It is responsible for analysing the different streams of sensor data and for identifying potentially hazardous situations. Several helpers are required that implement important auxiliary functions such as recording the flight data and giving feedback to the crew via a thin client interface based on HTML.

Programming Language

A real-time version of *Oberon* will be used as the programming language of choice. Oberon is a descendant of *Pascal* and was originally designed by N. Wirth in the late 1980s at the ETH Zürich. Simplicity in combination with full type safety makes this language superior to other languages such as the C-family and perfectly suitable for the programming of safety-critical systems. Another significant advantage of our choice is the "malleability" of the language, coming as an immediate benefit from our complete control of both the compiler and the runtime. For example, constructs for the support of ONBASS-specific issues such as fault tolerance, reconfiguration capabilities and real time control may be added freely on demand at any later stage.

As a starting point, *Oberon SA* will be used. This is a simplified version of the traditional Oberon language plus embedded systems support. It cannot be emphasized enough again that in sharp contrast to the usual C-based low-level systems programming culture no "unsafe" features are supported by Oberon SA, and the virtues of strong type safety and early detection of programming errors at compile time are maintained uncompromisingly. Most of the kinds of errors that typically lead to runtime catastrophes and that necessitate the use of complex static analyzers or model checkers are a conceptual non-issue in Oberon SA programs. In particular, Oberon SA does not support "inline assembler" code. Instead, some carefully selected low-level features like *interrupt procedures* and *register access* are built into the language in a clear and secure way.

It may be worth noting at this point that the choice of Oberon as an implementation language in ONBASS influences developers of both system software and application software for the onboard computer (including gathering and evaluation of flight data) but not or less so developers of auxiliary software such as development tools and simulators.

Runtime Platform

Outmost care will be taken in the design and the implementation of the ONBASS runtime to keep it simple, reliable and resource-efficient, while still maintaining a rich and abstract programming model and some advanced runtime "qualities of service" such as global memory management and real-time guarantees.

While most of the requirements to be met by the ONBASS runtime software are known and well-defined at this time, some open questions regarding the final underlying hardware still remain. For example, the redundancy patterns of the critical hardware components must be known very precisely before the corresponding runtime support of fault tolerance can be designed, let alone implemented. With the goal of flexibility and extensibility at a later stage, the ONBASS runtime architecture strives for most possible separation and encapsulation of hardware-specific aspects behind module boundaries.

Runtime Model

This section explains the onboard computer runtime from a dynamic, process-oriented perspective. The runtime will be designed to automatically launch a *boot phase* either after manual system reset or on programmed reboot request, and then to enter a (potentially) endless *main runtime loop*. Two boot modes will be supported: a) *normal* and b) *debug/ recovery*. The normal boot procedure incorporates the following steps:

1. Initialize and check the CPU
2. Check the system ROM
3. Setup and check the memory system
4. Start web server for error output
5. Read system configuration from ROM
6. Configure the system according to the configuration
7. Test each newly configured subsystem

8. Perform recovery if the system was not shutdown properly (flag in flight data recorder)
9. Rollback to last consistent state
10. Continue data capturing and start the main operating loop
11. Start background tasks

The core part of the final ONBASS onboard computer hardware will consist of a fault tolerant CPU, a system ROM containing the executables plus a description of the *system configuration*, the (volatile) *main memory* RAM and the *flight data recorder* (FDM). These components must be properly initialised and checked immediately after powering up the system. If necessary, reconfiguration steps must be performed to ensure the correct functioning of the core system. After this, the *configuration file* (in XML form) is read from the system ROM and the onboard system is configured accordingly. This includes the initialisation of input/output lines, starting helper threads and background tasks, and entering the main loop. Typical background tasks are *flight data recording*, *integrity checking*, *flushing* of the log file, and *launching* a command console application.

If normal boot fails, for example due to a corrupted configuration file, the system automatically performs a reset operation and enters *reboot*. After three failing attempts of booting normally, the *debug/ recovery boot mode* is entered. In this mode, only the absolutely essential subsystems are booted, for example CPU, ROM, RAM and a tool listening for commands on some predefined UART port.

The runtime activities are controlled by one *main thread* plus several *helper threads* running quasi-concurrently. In particular, a separate thread is associated with each *analyzer/ evaluator* of flight data. Other helper threads are responsible for auxiliary tasks like:

- Liveness checker
- Hardware checker
- Log writer
- Flight data recorder
- Web server
- Command tool

The purpose of the *liveness checker* or "*watchdog"* is monitoring the state of execution and in particular the detection of "wedging", that is, the endless execution of an inner loop. The aims of the *hardware checker* thread are checking the proper functioning of the hardware and keeping the system alive, even if hardware errors or deficiencies are detected. This thread runs periodically every 100 ms or so (in contrast to continuously) and at a high priority. If necessary, it performs reconfiguration actions. The *log writer* is responsible for periodically flushing the log file to the flight data memory (FDM). As alpha particles can cause FDM errors, it is necessary to scan the entire memory on a regular basis and, if necessary, to recover corrupted data. The danger of data loss will be reduced to a minimum by redundantly spreading the flight data over different flash memory chips in a way that ensures lossless recovery even in the case of a complete failure of one flash memory chip. The *web server* thread will be responsible for periodically reporting the current state of the airplane and sending an advice profile to the flight crew in a visual and humanly readable form. The use of the web server model allows basically any thin client such as a PDA to act as a user interface (HMI) device for the onboard computer. The *command tool* provides an option for interactive debugging and system monitoring via serial line.

The *main thread* controls the main system loop. This is a rough description of its actions:

LOOP
  Synchronise to the application 'heartbeat' clock;

Poll data ports and get data;
Record the data in the FDM and keep it cashed in RAM;
Detect the current flight mode;
Reactivate flight data analyzers/ evaluators;
Generate updated profile of the current airplane/ flight state;
In case of diagnosis or prognosis of fault(s) generate corresponding advice profile;
Update the display and, if necessary, alert the crew;
Update the system log and save roll-forward state marker
END

Note that, alternatively, flight data might be gathered via asynchronous interrupts.

Application Plug-In Framework

The ONBASS application framework will provide a standardized *plug-in mechanism* for application software and in particular for flight data analyzer/ evaluator logic, with the aims of simplifying application programming and enhancing modularity and extensibility. The basic idea is that evaluators of flight data will be separately compiled modules, will run under control of separate threads, and will follow a strictly event-controlled activation pattern. Also, access to current and past flight data will be simplified via a rich and abstract API. The application plug-in framework is targeted at enabling a wide range of experts who need not be computer scientists to write application software for ONBASS.

Flight Data Recorder

The task of the flight data recorder is to permanently and reliably store the entire flight data collected during each flight from different sensors at a rate of 1/ sec or 8/ sec. In the case of an accident, the data gathered during the last few seconds before the final impact is typically crucial for reconstructing the accident. Therefore, it is important to store the flight data samples immediately after they arrive, and the block caching/ flushing metaphor is not an option. In other words, the system should act as a "write-through" cache rather than as a "write-back" cache.

Flash memory as it is typically used for flight data recorders has several limiting constraints that must be taken into consideration in the design of a data structure that is suitable for recording flight data. For example, flash memory is typically partitioned into blocks of equal size, which are again partitioned in pages. While a page defines the smallest amount of data that can be stored per write-access, a block is the smallest amount of memory that can be erased. Because writing new data to a page presupposes erasing the old data, and because the total number of erase-operations performed on any block of flash memory during its lifetime is limited, write-operations must be distributed as evenly as possible across the flash memory. Also, unnecessary write operations must be omitted. Therefore, organisation schemes that imply frequent access to the same data block (for example to blocks containing metadata) as used in file systems such as *ext2* or *NTFS* are unsuitable. Several schemes were proposed, all of which are based on two principles a) copying the data to be updated to a new block and marking the old block "dirty" and b) garbage-collecting the dirty blocks to regain space. Unfortunately, such strategies require the system to keep a copy of the metadata in main memory and are therefore unsuitable in ONBASS because of the strict memory constraints. A different solution must therefore be developed.

Our idea is based on allocating files *statically*, with an option for later reconfiguration and reallocation. Because flight data records are of constant size they can be organized in the form of index-sequential files with fast access to individual instances. Note that the format of the flight data records does not need to be universally constant but might well be described by metadata as part of the system configuration.

In order to guarantee the integrity and correctness of the data in case of a failure or malfunction of the flash memory, an error detection/ correction scheme will be implemented. Physically the flight data

memory will consist of two flash memory chips, each of them containing exactly the same data. A Hamming or CRC32 coding scheme will be used to detect and correct errors. A background task constantly scans the whole flash memory for errors and re-establishes integrity by either using the redundancy provided by the Hamming coding scheme or by reading the correct data from the second flash memory.

2.3.3 Software Development

The software development in the frame of this project was performed under the lead of ETHZ.

First of all, the Minos OS had to be ported to the ETH hardware, which required changes mainly to the boot process on the target side. On the host (Active Oberon System) side, all existing Minos host application were ported, enhanced and smoothly integrated into the Aos environment. This embraced the extension and stabilization of the host terminal, various fixes regarding the interplay of the host system and the implementation of some helper tools.

On the programming language side, case studies (consisting of typical tasks such as the implementation of drivers) regarding use of certain language constructs have led to the definition of a minimal custom programming language for Minos Oberon. Enhancements and fixes to the Minos compiler and linker have been carried out to increase the overall platform stability.

The Minos Oberon system has been strongly modularized and was heavily extended, as a subsystem for each hardware component such as UART, SPI and MMC including partition handling and a block device interface were introduced. A dynamic plug-in concept has been established for the flexible integration of drivers into the system.

A small TCP/IP stack was written, which is used for the communication with the user interface. Parts of the PASS algorithm introduced by Londonmet have been developed: flight data acquisition and flight mode detection. For this purpose the maximal used flight data sets have been defined and an according software representation has been realized. The data acquisition protocol has also been designed and implemented and parsers for GPS, ADC and AE data over serial lines have been implemented as state machines. The system can be configured with an XML file. An appropriate XML parser has also been implemented.

In order to support distributed development, a source code repository based on subversion has been set up at ETHZ.

By the end 2006, based on the overall bad condition of the used compiler, it was decided at ETHZ to ask Prof. N. Wirth to develop a new version of the compiler. Dependent components such as the linker and loader were also adapted. Minos was adapted accordingly.

The compiler was finished in mid 2007. The testing framework developed in collaboration with Robinsons in 2006/2007 was heavily used to verify the correct behaviour of the new compiler.

In addition, support for the IrocTech FPU hardware was added to the compiler. Benchmarks indicated a significant speed increase by using the hardware FPU in comparison to the standard software FPU emulation.

The Minos Operating System was ported to the fault tolerant IrocTech hardware which required main changes in the following components: Bootstrapping, UART driver, MMC driver and the OS timer. Various other components were affected to enable the full potential of the system.

A fault tolerant flight data memory subsystem based on flash memory was implemented. Very high reliability could be achieved, by combining physical data duplication and fault detection by CRC-32.

In parallel, ETG concentrated on the design and development of the Lightweight ONBASS User Interface Server (LOUIS). LOUIS is the interface between the ONBASS system and the user terminal, which is a PDA running the Microsoft Internet Explorer. It is designed as a kind of web server. It uses the web protocol HTTP and provides dynamically created HTML web pages containing the information for the pilot. The design of the dynamically created web pages was provided by SPIRIT (ORATION). The used low level protocols TCP/IP and SLIP were implemented by ETHZ on the target system. A respective software version was provided to ETHZ and successfully integrated into the overall ONBASS software package.

An internal deliverable (ONBASS 'User Manual') detailing the operational features of the system from the aspect of the user's interaction with the ONBASS User Interface was prepared and distributed during the course of the project.

2.4 Hardware Development

2.4.1 Objectives

The main objective of this work package consisted in developing a highly reliable system including fault tolerant processors, system memory, flight memory and communication interfaces. This work package also included the proof of the fault tolerant mechanism. The reliability of the system hardware has been proven analytically and practically.

2.4.2 Hardware Structure Definition

The ONBASS device must be able to interpret input data, store required information for analysis during flight and after flight, present useful information to the user and allow off-line access to stored data. Special emphasis is put on the fault tolerant aspects of the product. The following key features of the device can be summarized:

- Requires a power supply with a nominal voltage of 27.5 Vdc or 13.8 Vdc (manually selectable)
- Integrates a backup battery pack allowing for 10 minutes of functioning without service interruption. The device will manage the recharge of the backup battery
- Power ON/OFF button
- Built In Self Test (BIT) button
- An RS232 serial data link (2400 to 9600 baud) for navigation (GPS) systems
- ARINC 407 3-wire XYZ synchro heading input or 4-wire DC Sin/Cos selected course drive inputs
- An RS232 serial data link (2400 to 19200 bauds) to a computer (PDA or Laptop) for a Human to Machine Interface (HMI) to be used on-line
- A general–purpose USB connection that can be used as needed
- An RS232 serial input (1200 or 9600 bauds) for a Air Data Computer (ADC)
- An RS232 serial input (1200 bauds) for the Altitude Encoder
- A set of LED indicators for the health state of the device and plane
- An extension connector for accessing for interfaces to external inputs via memory mapped I/O
- Fault-tolerant CPU to:
    - analyze the input data
    - present relevant information to the user, through a HMI
    - store data in the Flight Data Memory (FDM)
- Fault-tolerant system SRAM & FLASH memory for the CPU usage
- Supports off-line access to the system FLASH for the update of the internal program
- Non-volatile FLASH memory for the FDM unit, using standard FLASH cards
- Supports off-line access to the FDM unit for downloading data to a diagnostic equipment via USB
- Supports extraction of the FDM FLASH memory cards in case of an accident
- Proposes equipment to allow reading of the FDM FLASH cards outside of the device
- Physically places FDM cards in a hardened case for a better protection during an accident
- Hardware watch-dog for monitoring the activity of the CPU and the voltage levels in case of a latch-up
- Presents Points of Tests to allow production testing of the principal modules of the device
- Supports off-line Built In Self Testing (BIST) of the device

In the frame of the ONBASS project two hardware prototypes were developed:

- The ONBASS prototype hardware and the
- ONBASS Fault-tolerant hardware

ONBASS Prototype Hardware

ONBASS Fault-tolerant Hardware

ETHZ took over responsibility for the prototype hardware that was built around a conventional, commercial processor, not considering the aspects of hardware fault-tolerance. The prototype hardware was expected to serve as a platform for developing the necessary software in due time before the final ONBASS hardware was expected to be available. Figures 2.14 and 2.15 depict the respective hardware architecture.

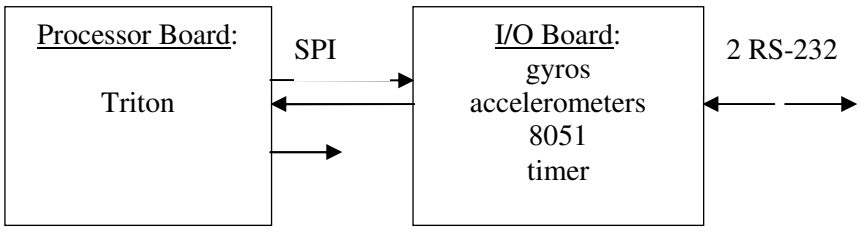


Figure 2.14: ONBASS Prototype Computer

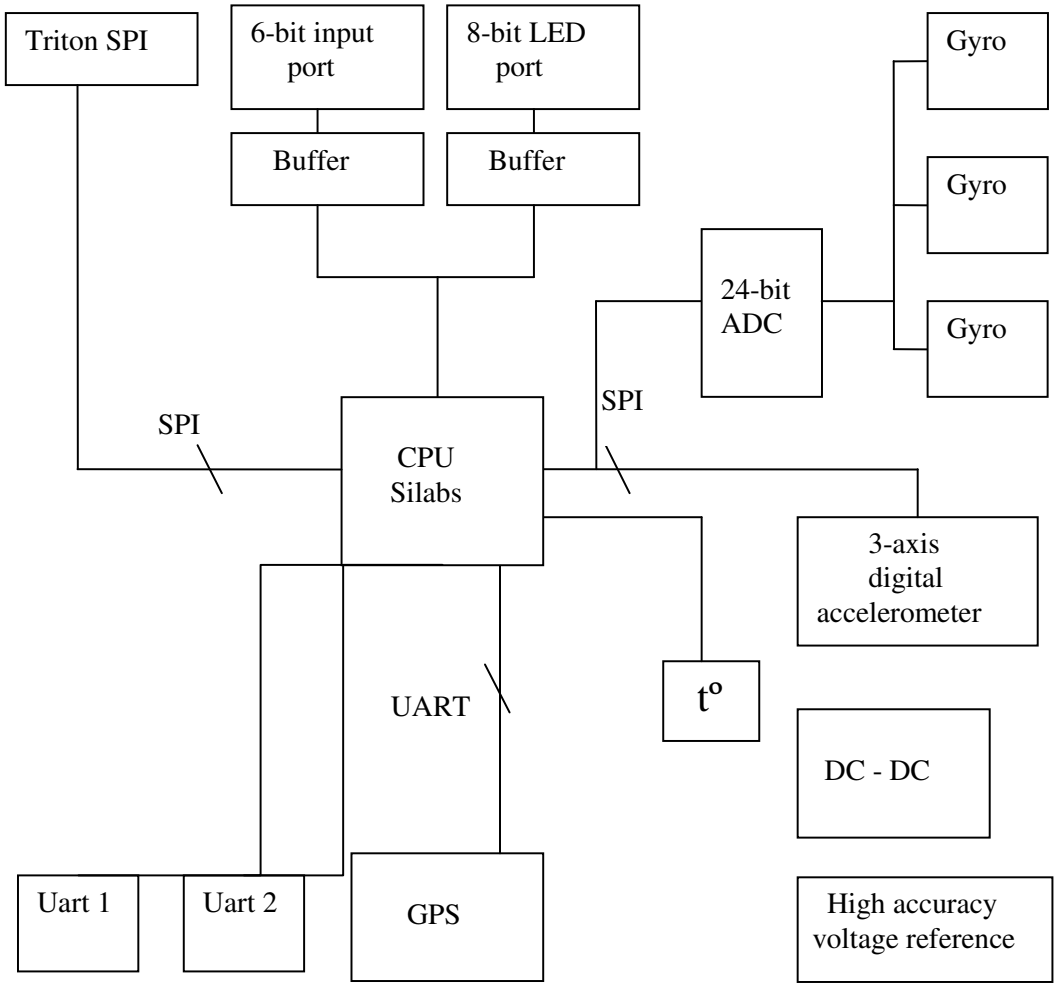


Figure 2.15: ONBASS Prototype Computer – I/O Board

The fault-tolerant ONBASS hardware was developed by IrocTech. The respective block diagram is shown in figure 2.16.

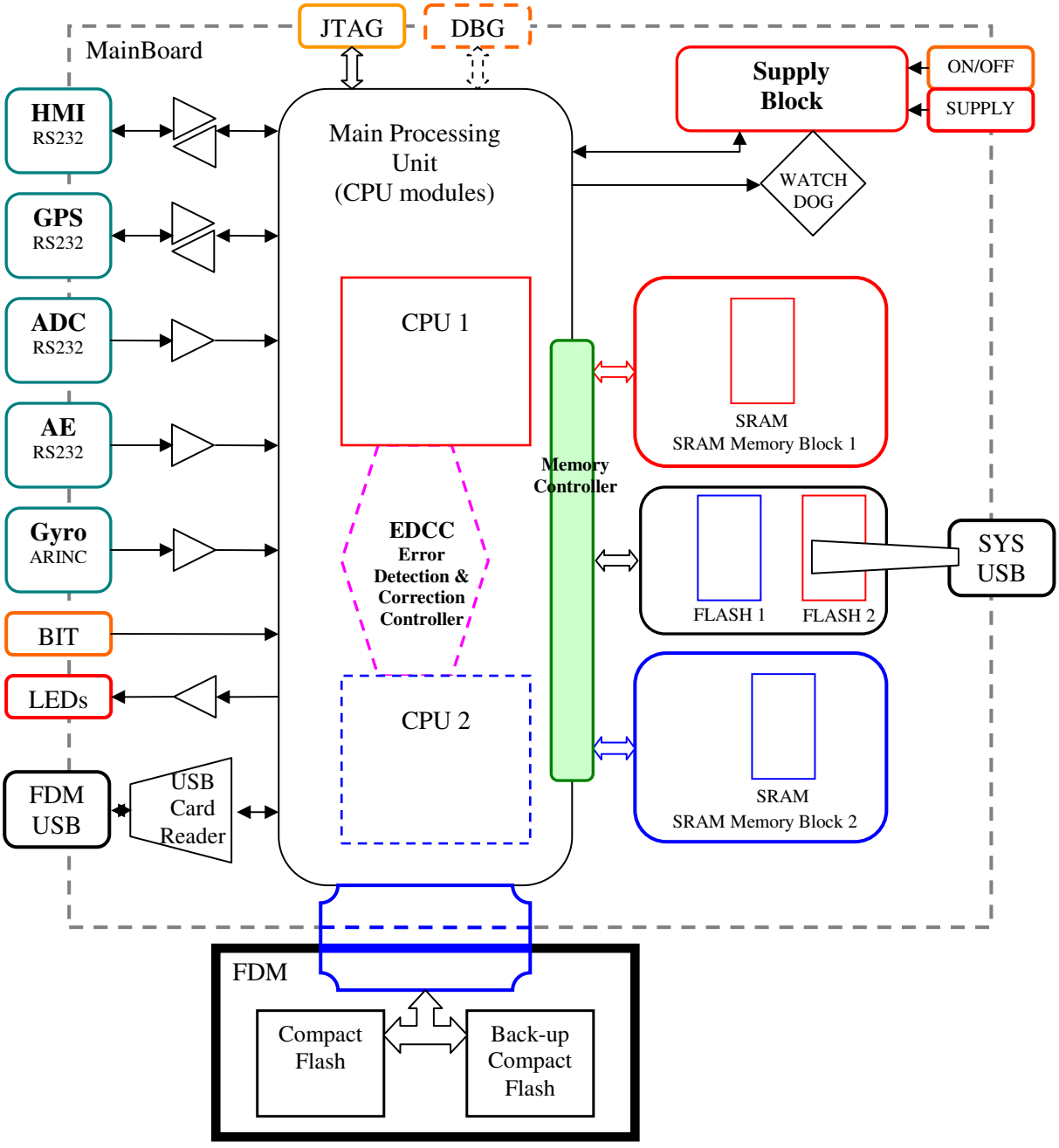


Figure 2.16: Block Diagram of ONBASS Unit

2.4.3 Hardware Development and Verification

Both ONBASS hardware platforms were already developed in 2006. The final version of the prototype hardware was delivered to the ONBASS consortium for subsequent software/hardware integration on December 12, 2006. The fault tolerant hardware was finally tested and verified at the beginning of 2007 and was delivered to the ONBASS consortium for subsequent software/hardware integration in March, 2007. It should be noted that after delivery of the equipment still some firmware updates were made in order to correct some minor bugs found during software/hardware integration.

In the following some more details are given regarding the functional testing of the fault-tolerant hardware. The tests conducted on the hardware can be divided into functional tests and fault-tolerance tests.

The functional tests aim at the verification of correct functioning of the different modules integrated on the board.

System SRAM memories

The processor can access the SRAM memories correctly. Different data patterns are written and read from the SRAM memories. The tests cover all the addressing space of these memories. The processor can do the following types of operations properly:

- Write any data to any word in the memory addressing space.
- Read any word in the memory addressing space.

Different data patterns have been written and read successfully from the SRAM memories. The memory blocks and the interface between the memory and the CPU work correctly for all addressing modes of the CPU (byte/halfword/word) and all types of operation sequences. Validation has been carried out at the nominal working frequency of the CPU, without the need for wait states.

System FLASH memories

A specific interface has been developed between the CPU and the FLASH memories. This interface allows the implementation of the various operations allowed by the FLASH memory. In particular, the CPU can:

- Lock/Unlock blocks for protecting the memory against accidental writing
- Erase block(s) in order to prepare for write
- Write any data to any word in the memory addressing space.
- Read any word in the memory addressing space.

The interface with the System FLASH memory works at the maximum working frequency described in the memory specifications.

CPU JTAG Programming & Debug Interface

The ONBASS board also offers a debug interface that allows the software programmer to access and control the embedded CPU. Through the intermediary of this interface, we can also access all the processor memory space, including System SRAM, System FLASH, serial ports, FDM memories, etc. This interface has been validated and works correctly.

Serial connections

The following validations have been carried out:

- The processor can correctly send and receive data through the different serial connections
- The connection can be made at various baud-rates specified in the ONBASS System Specifications document

USB connection to Flight Data Memories

A specialised SD card reader takes care of the connection between the SD card and the USB connection of the ONBASS unit. Using this feature, the operator can use a standard PC to access data stored in the FDM blocks.

The USB connection to the FDM blocks has been validated and works at USB2 speeds. The SD cards can be correctly accessed at speeds of up to 6 MB/s.

The access of the SD card reader is controlled by the CPU. During normal functioning, the CPU accesses the FDM for storing flight-related data. The SD card reader is connected to the FDM only during data retrieval sessions, when the operator connects to the ONBASS unit using a PC and downloads flight data from the SD cards.

ARINC407

The interface between the ARINC interface circuits and the CPU has been developed and works correctly. Unfortunately, we could not validate the correct treatment of the ARINC signals as we could not yet find an ARINC generator. However, The ARINC interface circuits are Application-Specific ICs (ASICs), specially designed for the acquisition of the ARINC Synchro/Resolver signals. Thus, it should work without problems. We will validate the functioning of this module as soon as possible.

Real Time Clock

The real time clock has been validated. The processor can set the time into the RTC component and can read time from it correctly. The RTC has a lithium battery with an estimated lifetime of several years.

Power Supply

The power levels have been validated on the ONBASS board. The following checks have been done:

- The 3.3V is generated correctly for the SRAM memories, the FLASH memories, the FDM memories and the IO's of the FPGA.
- The 1.5V of the FPGA core is generated correctly.
- The 5V of the IO's connectors is generated correctly.
- All the voltage levels above are stable when the input power changes between 14 and 28V (according to the specifications). Additionally, the power block is able to function correctly within an extended voltage range of 10-38 V.

Power backup (battery)

The power backup has been validated. The following checks have been done:

- The device automatically switches to battery power when the main power is off
- All voltage levels are stable during the transition between main and battery power and until the battery wears off (less than 5% remaining power)
- The battery starts recharging as soon as the main power is reconnected.

The battery will last from 2 to 3 hours when the main power fails.

Hard errors tolerance testing

When one block of the SRAM memories is defective, the system will continue working properly by retrieving the data in the second memory group (the system SRAM memory is duplicated). This was validated by cutting the power of one memory group and checking that the processor continues to read the correct data from the second memory group. This procedure has been applied for the two memory groups. This protection is effective against both individual defects in the memory chips (such as stuck

bits) and also total chip failures (one or several memory chips from the same memory block don't work at all).

When one block of the FLASH memories fails, the system should continue to boot properly by taking the data in the second memory group (the system FLASH memory is duplicated). This was validated by cutting the power of one memory group and checking that the processor boots normally from the second memory group. This procedure has been applied for the two memory groups. This protection is effective against both individual defects in the memory chips (such as stuck bits) and also total chip failures (one or several memory chips from the same memory block don't work at all).

The double rail power supply has been validated during the validation of the fault tolerance of the system memories. The processor can turn on/off any of the DC-DC converters on the board. This was used to simulate failures on the power components. When a power rail fails, the dual rail continues functioning and thus the system works normally.

Soft Errors tolerance testing

The System SRAM memories are protected from soft errors by error detection and correction code. When an error is detected after a read from the memory, the error is corrected and the correct data is rewritten to the memory.

The error mitigation algorithm detects and corrects single errors and can also detect double errors. In any case, if an error is undetected by the correction mechanism (very improbable case), it will be detected and corrected by comparing the outputs of the two memory blocks.

The implemented error detection and correction mechanism is validated by injecting errors into specific locations in the memory and checking that correction is taking place and the stored program continues to run properly.

The System SRAM memories are protected from soft errors by comparing the data during read with the data from the dual block. In case of errors, the content of a mirrored, inverted firmware image is used for deciding the correct data.

At power up the program stored in the FLASH will be copied into the SRAM memories. During copying data, the errors (if any) will be detected and the correct version of data is written to the SRAM. This mechanism is validated by injecting errors into the FLASH memories and checking that the data written into the SRAM is the corrected data.

2.5 Integration, Verification and Demonstration

2.5.1 Objectives

The objective of this work package was to integrate the verified software modules on the respective hardware prototypes and to perform an overall system verification making use of suitable simulation and laboratory set-ups. Once the overall system performance was verified, the demonstrator was demonstrated in real flights in November 2007.

2.5.2 Integration

Since the ONBASS prototype hardware was already available since summer 2006, the integration of the ONBASS software started at this point of time in parallel to the software development. At the very end the following components were successfully integrated and verified on module level on the prototype hardware:

- Host System:
  - Minos Terminal
  - Minos Oberon language report
  - Compiler
  - Bootlinker
  - Module definition browser
  - Module decoder
- Operating System
  - Boot loader
  - Kernel, platform specific
  - Memory management
  - Floating point emulation
  - Debug output
  - Low level debug output
  - File system
  - Dynamic module loader
  - Operating system interface
  - Scheduler
  - Generic device interface
  - Uart driver
  - Power management
  - Block device interface
  - RamDisk
  - RomDisk
  - Block Device Volumes
  - Block Device Cache
  - SSP driver
  - IOBoard driver and controller
  - MMC driver and controller
  - flight data file system
  - TCP/IP
  - SLIP

- Applications
  - Webserver
  - Data Types and DataFrames
  - Configurable Output
  - String Scanners
  - Data Acquisition: DataRouting
  - DataAcquisition: Parser Plugins
  - DataProcessing: FrameCapture
  - DataProcessing: FlightModeDetector
  - DataProcessing: FrameStorage
  - Configuration
  - XML parser
  - Fault injection
  - PASS implementation

The fault tolerant hardware was available since March 2007. The software integration on the fault-tolerant hardware was, therefore, based on the already reached integration status of the prototype hardware. This led to an accelerated integration process on the prototype hardware.

The main effort of this integration was put on the operating system related items like platform specific kernels, memory management or floating point emulation.

Since the host system and the application are platform independent, the whole application part ran without modification on the fault tolerant hardware as well, as the language guarantees platform independence. All modules that are not platform independent are marked as unsafe by the import of the pseudo module SYSTEM.

Finally, the whole port to the fault tolerant platform was empirically tested.

2.5.3 Verification

For the Factory Acceptance Test, the system has been verified using a variety of simulation tools set up in a laboratory environment. The configuration used is illustrated in Figure 2.17 below.

The bottom PC is used to run WinAOS and the Minos development system. The FPGA logic gate configuration is loaded from the PC to the FPGA using the ACTEL FlashPro and SoftConsole tools over the USB link to the ONBASS unit. This is used to download the ARM processor logic configuration and hardware level fault tolerance logic to the FPGA. The Minos Terminal communicates with the Minos Boot Loader in the ONBASS unit and is used for downloading the ONBASS software, controlling its operation and for debugging operations. In this case the ONBASS Web Client is shown on this PC, it could be on another PC or on a PDA, indeed any browser that supports the SLIP web protocol, it is connected via an interface on the ONBASS unit (possibly to USB on the PC via an RS232 adapter).

The top PC is emulating the aircraft and airfield using the Microsoft Flight Simulator. It outputs flight information via a driver to the ETG Flight Data Interface. This can emulate an Air Data Computer or GPS data format on the RS232 link to the ONBASS unit. In this way the unit ‘thinks’ it is receiving data from real aircraft equipment. It is possible to run several instances of the Flight Data Interface, one for each piece of equipment to be emulated on the aircraft. A joystick is used to control the aircraft.

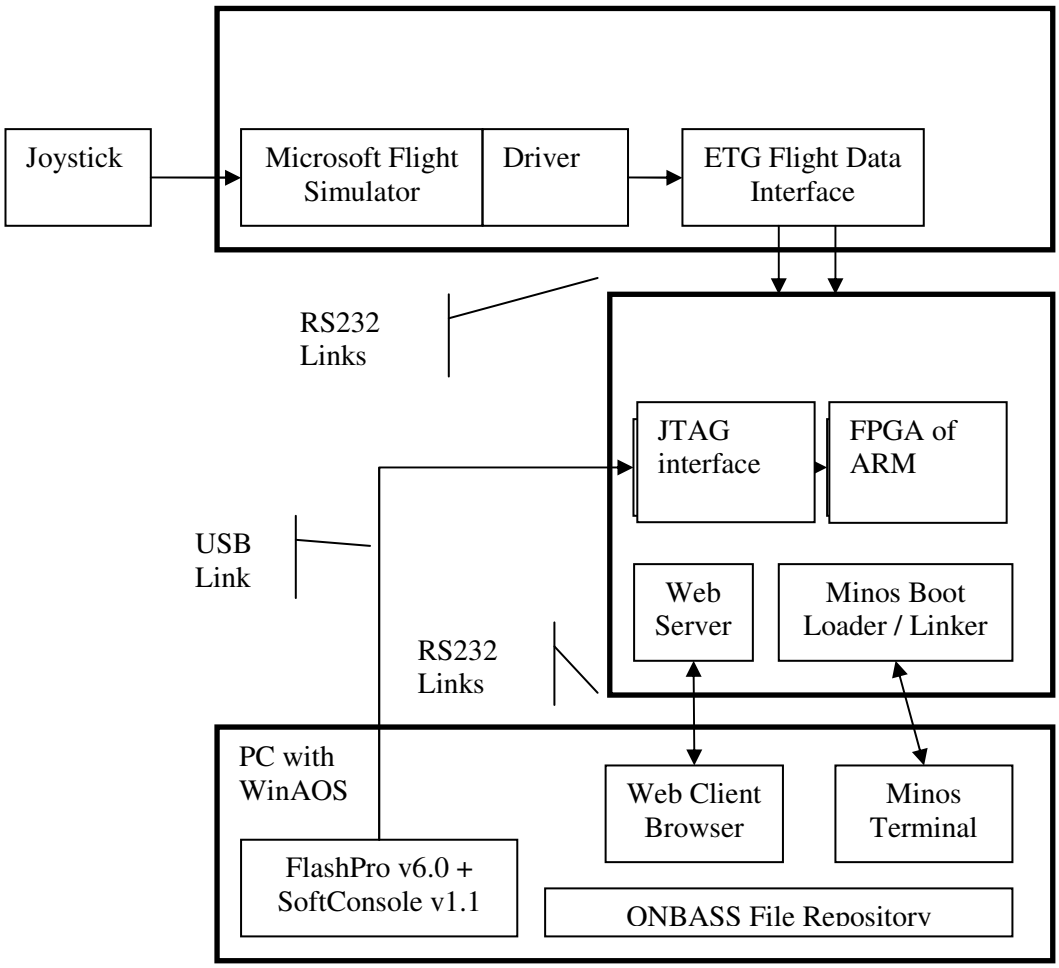


Figure 2.17: Setup for Factory Acceptance Testing

The set-up sequence was as follows:

- The FPGA logic gate configuration was developed by the hardware engineering team and stored in the ONBASS File Repository.
- The FPGA image was downloaded to the ONBASS unit and programmed into the FPGA.
- The ONBASS software image was prepared using WinAOS (middle click on MinosBuilder.call).
- This initial linked image was downloaded to the ONBASS units RAM memory via the SoftConsole tool via the USB interface and the JTAG unit on the ONBASS unit.
- The Minos Terminal was now used to command the Minos boot linker on the ONBASS unit to burn the program image into flash memory for permanent storage, using the command Minos.Burncore.
- The ONBASS has now been configured in terms of both hardware and software.
- The Microsoft Flight Simulator was installed on a PC.
- The MFS drivers were installed on the PC (files FSUIPC.dll and .ini).
- The Flight Data Interface was started and configured for the device for which it is emulating the data format.
- The internet connection was set up on the PC that is hosting the WEB Client for the ONBASS HMI.

The whole system was now operational.

The tests were separated into logical groupings which, where possible, followed the original requirements set out in the System and Architecture Specification. The purpose of the FAT was to verify that the original requirements defined have been covered by the implementation of the system. Every requirement has been reviewed and where appropriate verified. Each requirement is reproduced directly from the system specification followed by the verification / test result:

- **Findings:** Describe the test result, and details of the source of the test or test information if it is not self evident.
- **Verified:** Indicates the result of the verification / test, either Yes or No, followed by the initials of the person(s) who have done the work.
- **Observations:** Describes the context, provides clarification and if needed the relevance of the verified field.

The following test methods have been used:

- Code Inspection
- Document Inspection
- Hardware testing, to test parts of the hardware independently
- Module testing to test parts of the software independently
- System tests to test and verify the whole integrated system

A total of 181 tests were conducted with 171 tests passed. The failed tests did not affect the functionality of the system needed for the flight tests. The detailed results of the system verification were summarised in the respective deliverable report.

2.5.4 Demonstration

The real flight trials were carried-out taking of from the Wallmuehle airport in Straubing, Germany. All test flights were concluded by returning to the same airport. The reason for using this airport was that on the one hand it is the base for Avionik Straubing GmbH which provided the Piper Cherokee Lance aircraft used, while on the other hand, it is a typical European GA airport, permitting thus the maximum degree of realism in the demonstration activities.

As part of the test flights' demonstration activities, various consortium partners were involved bearing different responsibilities in the whole process. ETG were responsible for arranging for the aircraft to be available and for installing and connecting the system on-board. Representatives from ETG and ETHZ were responsible for going through and reviewing the previously prepared flight scenarios with the test flights' pilot and preparing/arranging for the appropriate experimental set-up or adjustments to that. Then, during the flights themselves, they were responsible for recording the whole procedure and for registering the associated results. The whole procedure of carrying out the (real) flight trials involved the following sequential steps:

1. Installation of ONBASS in the aircraft.
2. Review of proposed Flight Scenarios with Pilot(s) regarding practicality, safety and sequencing for flights.
3. Make corrections and/or adaptations to the flight scenarios.
4. Allocate flight scenarios to Flight Plans with Pilots.
5. Formalise Flight Plans (Pilots).
6. FOR each flight plan of scenarios:
   - Conduct the ONBASS equipment pre-flight checks (on, connected, receiving ADC data, enough MMC disc space, browser window for HMI visible, sounder working …)
   - Ensure ONBASS logging and storing.

- Carry-out the flight(s) with an ONBASS operator on-board.
- After the flight, save the ONBASS log and the flight data memory to archive storage.

7. Do the post flight analysis of the FDM and log to verify the compliance of the actual system behaviour with respect to the expected behaviour and what actually happened in the flight (from operators logbook).

On board the aircraft for these trials, the ONBASS system interfaced with:

- An Air-Data Computer (ADC-200) - via an RS232 serial link
- A Garmin GPS295 GPS device - via an RS232 serial link
- The aircraft's electrical power

The ONBASS unit was further connected to the ONBASS User Interface (i.e. a standard Laptop – alternatively could have been a PDA device, as illustrated below), via a USB or RS232 serial interface. The lay-out of the experimental set-up as well as the aircraft used for the flight trials conducted are illustrated in the following figures:

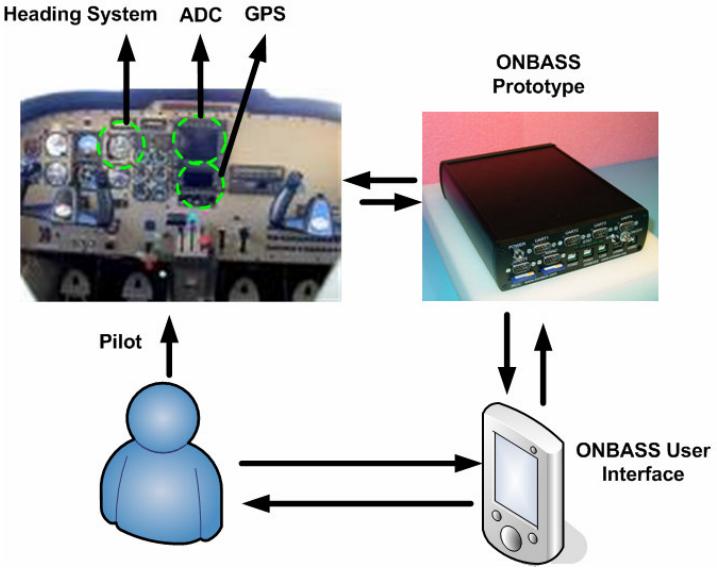


Figure 2.18: Real Flight Trials Set-up

Figure 2.19: A Piper Cherokee Lance (PA-32R-300)

A first integration test was performed on October 30, 2007 at the aircraft. The following problems were detected and solved:

1. The Air Data Computer provided data in a different format than expected -> after consultation with the manufacturer the correct protocol could be selected at the Air Data Computer (S format)
2. GPS data were not provided by the Air Data Computer -> the decision was made to use a direct connection with the GPS unit
3. A direct connection was established with the Garmin GPS 430. It turned out, however, that the data were not provided in the NMEA format, but only in the Garmin aviation format.
4. For that reason finally a Garmin GPS295 was connected providing data over NMEA

The integration test was finally concluded with a recording of data from the Air Data Computer during taxi operations.

The recorded data were further analysed in the laboratory before the start of the flight tests.

The following data were finally received:

From Air Data Computer:
- Indicated Airspeed
- True Airspeed
- Mach Speed
- Pressure Altitude
- Density Altitude
- Outside Air Temperature
- True Air Temperature
- Rate of Turn
- Vertical Speed
- Heading

From GPS:
- UTC Time
- Date
- Latitude
- Longitude
- Ground Speed
- GPS Height

Finally, three demonstration flights were performed on the 14th and 15th of November 2007.

2.5.5 Evaluation

In general, it can be summarised that the ONBASS System could be successfully demonstrated in flight. The trials carried out allowed a number of very useful conclusions to be drawn with regard to the usefulness/appropriateness of the system and its implemented checks, as well as with respect to any further future developments that the system may undergo.

Specifically, the following most notable conclusions were drawn as a result of this exercise:

- The system indicated a high degree of potential with respect to enhancing aviation safety, initially -in this case- in GA, but later also possibly in CA, MA or even helicopters. The main limitation to this potential offered by the system has been deduced to be the number of digitally available data signals on-board.

- The capabilities of the system even at this early demonstrator phase are way above and beyond those of a typical FDR system. ONBASS not only records the available data, it has been demonstrated to process and react to it in real time, including as part of a real flight.
- The design of the system is such that it is relatively straight forward to customise the safety checks included on a per aircraft case, as well as to 'on-the-fly' update the assigned safety thresholds and rapidly upload them to the system (via the XML config file).
- It was demonstrated that it is quite simple to reset the system in-flight (e.g. so as to ensure full system recovery) and the current state and conditions will be rapidly reacquired.
- Subtle adjustments/tweaking/fine-tuning and even revisiting of some of the implemented safety checks is required so as to ensure that the maximum potential of the system's capabilities is exploited. Nuisance warnings will have to be avoided too. Such adjustments and modifications can be swiftly implemented thanks to the carefully realised design of the ONBASS system, via the updating of the XML config file.
- In certain cases further work is required on understanding and resolving the differences in the results achieved by the system during the real flight trials, when compared to the simulated ones.
- Further in-depth testing both via simulated and real flight trials is required so as to investigate and eliminate any further system bugs and/or inconsistencies.
- Monitoring of the pilot behaviour while flying the aircraft and the recognition of respective pilot errors is hard to achieve in General Aviation. The differences in pilot behaviour are so significant that a respective behaviour is estimated totally normal by one pilot, while another one would judge it as a dangerous pilot error. Therefore, further efforts are required to be spent in the direction of establishing a model of "standard" pilot behaviour and to define reasonable thresholds for any associated warnings and hints.

3 Dissemination and Use[2]

This section will attempt to describe how the ONBASS consortium composition was suited to the task of promoting and commercially exploiting the results of the project, initially at the European level.

3.1 Individual Partner Exploitation Plans

3.1.1 Euro-Telematik

Euro Telematik, as a company that specialises in the provision of avionics systems for General Aviation has a great interest to further advance the project results of ONBASS and turn them into a commercially available product. As ETG already has a wide clientele basis coming from the GA field, both in Europe and beyond, it would be quite straight-forward for the company to attempt to promote and capitalise on this basis in the form of achieving a considerable number of sales of an ONBASS system.

ETG however appreciates that the ONBASS prototype is still quite far from becoming a system which could be immediately sold on the GA market and thus plans to pursue further research with respect to transforming the ONBASS prototype into a commercial system, including addressing in depth issues relating to the certification of such as system. ETG is also interested in extending the results of the ONBASS project from the General Aviation field, to that of the Commercial Aviation and perhaps even to Helicopters.

Further, ETG plans to investigate how it could build on the knowledge acquired/created as part of the project so as to extend its product offering(s) and develop adjacent and/or peripheral systems and/or applications in the fields in which the company specialises, i.e. Aerospace and Telematics.

3.1.2 London Metropolitan University

LondonMet as the originator of the concept behind the ONBASS project has already from a long time ago progressed the theoretical foundation associated with this concept on a 'global' scale which considers the rest of the aviation segments involved, i.e. the Federal (administrative/aviation authorities) and the Ground. In addition, the original scope of the ONBASS project encompassed automated reaction/mitigation actions (so as to address any flight hazards) being implemented by the ONBASS system via the FMS (Flight Management System). It is hence LondonMet's demonstrated interest to further extent the ONBASS results to these research areas, not yet covered.

Further, as the department from LondonMet participating in the ONBASS project specialises in IT and Computing, during the course of the project, the focus of LondonMet's participation was in also defining a FT design and the associated mechanisms for the system's CPU, as well as the reliability and availability theory to support this. LondonMet is therefore also interested in pursuing further research in this direction, i.e. of FT CPU design and FT systems in general, where it could build on the results achieved as part of ONBASS.

The Department of Computing, Communications Technology and Mathematics (DCCTM) hence plans to offer a number of graduate and post-graduate research opportunities in the aforementioned directions to young students, thus further disseminating and building on the ONBASS project results.

[2] ***Knowledge:*** means the results, including information, whether or not they can be protected, arising from the *project* governed by this *contract*, as well as copyrights or rights pertaining to such results following applications for, or the issue of patents, designs, plant varieties, supplementary protection certificates or similar forms of protection (Article II.1.14 of the contract).

3.1.3 iRoC Technologies

iRoC Technologies was the sole developer of the ONBASS Prototype HW. As the design of the associated platform incorporates a number of FT features and a highly reliable architecture, there is great room for exploiting these 'assets' in many other similar or adjacent fields of application.

iRoC intends to further research and develop these architectures and FT components and utilise them in its product line. Also, iRoC is very interested in following-up the ONBASS project results and extending them through additional research projects in the field of aviation, a field which because of the extremely pressing and demanding requirements associated with the systems employed, offers the company the opportunity to develop further high potential FT and reliable components and integrated solutions which iRoC could commercially exploit both in aviation and many other technological areas.

3.1.4 Robinsons Associates

Robinsons' has devoted a large number of efforts in the direction of 'solidifying' the ONBASS concept, architecture as well as the theoretical foundations behind it, in conjunction with LondonMet. This demonstrates the belief of the company in both the potential of the ONBASS 'solution' as well as its applicability to a large number of fields of safety critical applications beyond aviation.

Robinsons' has thus been very active (again in conjunction with LondonMet) in patenting the 'methodology' and critical elements of the ONBASS solution with a view to fortifying its position with respect to the future utilisation of the project's results in both aviation and other potential application fields.

As with the other partners of ONBASS, the company is also highly interested in expanding the results of ONBASS to Commercial Aviation, this being a much larger market, and one that the ONBASS concept has even more potential for making a marked difference. In order to achieve this, Robinsons' in conjunction with several other partners of ONBASS is seeking to land further research work in the direction of building-on the ONBASS results and taking them even further.

3.1.5 ETH Zürich

ETH Zurich as part of the ONBASS project was the main developer of the prototype system's SW. The system SW and components developed were by the very nature of the Oberon language used, quite generic and thus reusable for many other applications. This in general is a sound philosophy of the Operating Systems Groups of ETH Zurich, to try and make use and build-on previous research and development work carried-out by the Group, as much as possible, in the current project work. Already this was the case in the ONBASS project where SW components such as the compiler etc were developed building-on previous research work.

Similarly it is the intention of the Group to make re-use of the ONBASS results in future work, be it follow-up projects to ONBASS, or work in similarly quite demanding (in terms of requirements) and critical (in terms of safety), application areas.

3.1.6 SPIRIT S.A.

In order to maximise the benefits arising from the exploitation of the ONBASS project results SPIRIT tried to couple as much as possible dissemination/marketing initiatives with exploitation activities. Further, primarily SPIRIT aims to exploit the technical results, experience and expertise gained from the company's participation in ONBASS so as to strengthen its current product offering, especially in the aerospace sector. Systems/technological solutions building on the ONBASS concept could much more efficiently be pursued by the company, either in the aerospace or other adjacent industrial domains.

As a partner in the ONBASS project SPIRIT will be in a position to further extend its corporate understanding of the relevant industry (aerospace) and to increase the company's presence in the related market. Further, it could then as a result extend its business activities to new (adjacent) markets and built-up and sell technical know-how in the form of consulting.

Ultimately however, the main objective of the exploitation plan of SPIRIT with respect to the ONBASS project was to aid in the development of an innovative, state-of-the-art product/solution (i.e. the ONBASS system), which could be exploitable both in the main target market as well as in other adjacent markets. As this was achieved, ONBASS exploitation will be shifted into a more 'commercial' natured campaign where the exploitable results could be numerous but the most prominent (and relevant to a private firm such as SPIRIT) being that associated with the collection of revenues from related sales, royalties and/or service charges.

As already described above, it was SPIRIT's belief that in order to achieve these ambitious exploitation objectives, dissemination initiatives would have to be efficiently integrated with exploitation activities and thus SPIRIT invested heavily in marketing/promotion of the project, its objectives, context and results, disseminated related information, thus establishing some basic 'sales channels' for future potential commercial exploitation, possibly pursued through a network of business alliances and contacts in the aerospace industry.

3.2 ONBASS Consortium Exploitation Plan

The ONBASS consortium brings together organizations from five different European countries and quite diverse technical knowledge areas which were to plan a detailed strategy to exploit the ONBASS results. The ONBASS consortium can still be considered to be in the initial phase of finalising this strategy; however, already some possibilities for potential commercial exploitation of the projects results have been identified.

With respect to the exploitation of the project results there are several options open to the consortium at present. The two main options open are: either the system as a whole could be promoted and sold on a Consortium basis (in this case the Consortium, as a single entity, would be responsible for the exploitation the outcomes of the project) or on a Partner basis (in this case each partner could exploit, according to the exploitation agreement, any outcomes of the project on an individual basis). In principle both models are considered by the partners, depending on the respective geographic markets addressed.

Regarding the associated geographical areas/markets identified, the ONBASS project outcomes/results could be exploited in individual countries, across Europe or internationally. At a first stage, a local distribution would have to be considered, starting with Germany, the United Kingdom, Switzerland, France and Greece, the countries from which the ONBASS consortium partners originate from. Nevertheless exploitation across Europe will be the medium-term objective of the consortium and also any opportunities for an international exploitation (outside Europe) will also be investigated.

In order to facilitate these actions, the consortium will jointly elaborate a more detailed Exploitation Agreement that will be put in effect by signing/affirming it, post-project. This agreement will include common exploitation rules, based on the ones that are defined in standard forms of the European Commission (e.g. the Project Contract and the Consortium Agreement), and will cover aspects like exploitation rights, distribution of revenues, sharing rules, IPRs, pricing policies, maintenance/support policies, possible upgrades, etc.

3.2.1 ONBASS Exploitation Lines & Scenarios

The ONBASS consortium even at this early stage of the more concerted exploitation activities has identified the main exploitation lines and scenarios to be pursued in the post-project period. These include:

- *Internal (to company) exploitation of results*, i.e. through the dissemination of the knowledge created during the course of the project within the individual partner organisation.
- *Internal (to consortium) exploitation of results*, i.e. through the dissemination of the knowledge created during the course of the project within the ONBASS consortium.
- *External exploitation of project results (knowledge or know-how obtained)*, i.e. by establishing business contacts and strengthening the presence of the consortium and its partners in the related industry.
- *Patenting*, i.e. by fortifying and safeguarding the knowledge created during the project for the purposes of enjoying the benefits arising from the generation of such innovative concepts.
- *Standardising*, i.e. by establishing the ONBASS philosophy as a standard norm, then a major step for the adoption of the system will have been made.
- *Commercialisation of system*, i.e. by translating the project results into product(s) and collecting the resulting revenues etc.
- *Preliminary market study*, i.e. by assessing the market status and particularities the consortium will be better equipped when beginning a full commercial campaign to promote and sell the system.

Some potential exploitation scenarios for the ONBASS project results have already been identified and include:

- Collective exploitation
  - By the Consortium as a single entity
- Individual exploitation
  - In any Country
  - By any Partner (individually)

3.2.2 Innovative Aspects of the System

The ONBASS system, by design, possesses a number of innovative features as far as its operational characteristics. These innovative features offer the users of the ONBASS system a series of benefits of which the most predominant are described in following:

- *Real-time active safety* – will greatly assist in reducing the number of general aviation (as the primary field of application) accidents and resulting casualties.
- *Fault-tolerant ONBASS system processor and RAM* – will result in extremely high reliability and availability of the system and very rare on-site maintenance actions.
- *Fault-tolerant ONBASS Flight Data Memory* – will result in a high integrity and trustworthiness of the stored data.
- *Resilient ONBASS Software core* - will greatly contribute in the extremely high reliability and availability of the system, as well as in the uninterrupted and efficient provision of the system's services.
- *Independent power supply* – will ensure that recording of crucial aircraft parameters continues even prior to/during a hazardous situation/accident or even in the event of the loss of aircraft power.

3.2.3 Initial Target Market

The initial target market for the ONBASS system will obviously have to be European GA aircraft. This is only reasonable as the geographic distribution and business 'reach' of the ONBASS partners mostly covers this continent. Further, setting-up distribution channels and building on the business contacts and partnerships of the partners would make potential sales in this region much more likely and straight-forward.

This initial target market, according to the data provided by IAOPA for 2003 is constituted (approximately) by some 46915 GA aircraft of various types. Adopting the US GA fleet growth figure of 0,5% per year, it is estimated that this initial target market will consist of some **52503** GA aircraft in Europe in 2008.

Having established a strong presence of the ONBASS system amongst the European GA and having demonstrated the benefits of adopting the system, it will be possible to expand the business operations of the consortium to the much larger US market and further the international market worldwide.

3.2.4 Typical Customer Profile

The ONBASS system by its very nature is not essential for flying the aircraft bearing it. It will however be highly desirable in order to make aircraft operations safer and more reliable. As such it will be interesting to a vast number of potential users/customers.

The profile of the potential customers of the ONBASS system mainly includes: general aviation aircraft owners; aviation clubs and flying schools; air-taxi companies; special services providers (e.g. fire-fighter, policing etc) and in general organisations operating general aviation aircraft.

The majority of these groups of potential customers of the ONBASS system (i.e. all those non-state owned organisations/bodies) have a limited purchasing capability. As the ONBASS system is not strictly necessary for piloting the aircraft, they will have to be convinced of the benefits of purchasing and installing the ONBASS system in their aircraft.

In addition, the ONBASS system will have to be available at a price that will be affordable to the potential users/customers of the system. An initial such estimate of the ONBASS consortium partners was that the system price should lie somewhere in the range of 5-10 k€. At the end of the project it was concluded that the system could be provided at a price of around 2000-2500€, i.e. well within the initial estimate of the partners and the purchasing capabilities of the system's potential customers.

3.2.5 Competition

Research into potential competitive solutions with respect to the ONBASS system, as carried-out by the ONBASS consortium, has verified the initial conception of the partners that there is no direct competition to the system in the global aviation market. The concept of active safety for aviation is one of the most innovative and technologically advanced solutions at present in the global aerospace industry.

3.2.6 System Supply and Support

There is only one way really that the ONBASS system could be supplied and that is as an all inclusive unit. Selling the individual parts/components would not make any sense as they are all interlinked in such a manner that a stand-alone component of the system would basically be useless. It should be noted however that the basic principles, concept and philosophy behind each ONBASS component are such that would be applicable in the design of many other stand-alone units of a series of adjacent markets. Such implementations could in future result in further revenues for the ONBASS consortium partners.

With respect to the support of the system, there are two aspects that should be considered. On the one hand the system should be upgradeable (in terms of software), e.g. to include further configuration parameters, aircraft types, sensor devices etc. This is foreseen to be possible for the users of the system against a small subscription fee (or on a case-by-case software upgrade at a higher cost). On the other hand, there are the rare cases (because of the fault-tolerant design of the system) where an on site maintenance of the system will be required to be carried-out to rectify any faults in the system, perhaps by replacing some hardware element. At present it is foreseen that it would not be cost-efficient for the ONBASS consortium partners to provide such support as on the one hand it is expected that these situations will be rare and on the other hand the geographically dispersed locations for providing such support would arise significant costs. Perhaps out-sourcing such work to local technical services organisations would be a more appropriate scenario.

3.3 ONBASS Exploitation Activities

During the course of the project there were 2 most notable activities which were undertaken with a view to the future exploitation of the ONBASS results. These were:

- LondonMet arranged for, attended, presented and discussed regarding the ONBASS project in Toulouse with the Airbus Safety Department, as the representative of the consortium. The Airbus people demonstrated a strong interest in the project which is to be followed up in the future.
- Additionaly, ETG & Robinsons attended and presented the ONBASS results (as representatives of the consortium) at a meeting with EASA in Cologne on 5/9/2007, investigating the future certification/commercial potential of the ONBASS system. Once more EASA demonstrated a strong interest in the project which is to be followed up in the future.

3.4 ONBASS Consortium Dissemination Plan

The ONBASS consortium had identified the strategy to be followed with respect to the dissemination of the project results, quite early on. This dissemination strategy was to be addressed in three (3) levels.

The 3 levels of the ONBASS dissemination were the ‘Dissemination Planning’, the ‘User Driven Dissemination’ and the ‘Research Driven Dissemination’. Each level addressed a correspondingly more ‘penetrating’ level of dissemination/promotion.

At the first level the dissemination activities were mostly concentrated on the development of the material to accompany the dissemination activities i.e. the Project Logo, the Project Brochure, the Project Presentation, the Project Website, the Project Newsletter(s), etc.

The second level dissemination activities would focus on the creation of awareness in relation to ONBASS, initially as far as the general public is concerned and then in a more targeted manner in the related community (i.e. the European aerospace and aviation industry). This was aimed to be achieved by a series of press releases (both printed and electronic in related media/press), the creation of a mailing list and the circulation of the project newsletters, the publication of partners’ or the consortium’s findings, the attendance and/or the presentation of the project objectives and results at workshops, seminars, conferences etc. In following the ONBASS project results would enter the pre-product launch phase.

Finally, at the third level (‘Research Driven Dissemination’) dissemination would encompass activities which were to be aimed at clustering the ONBASS project and its results with other research (whether EU co-funded or other) or even commercial/industrial efforts along the same lines so as to maximise the synergy in the field as well as to seek to exploit the potential of the ONBASS results within the

spectrum of a wider commercial/industrial effort. In addition, the ONBASS consortium was to seek to interact with the bodies and authorities of the industry with the aim of defining the way ahead, to enable technology transfer and communication with the EU R&D community etc. Following these activities the results of the ONBASS project would be ready to enter the 'Product launch' phase.

The three levels of ONBASS dissemination are illustrated in the figure below:

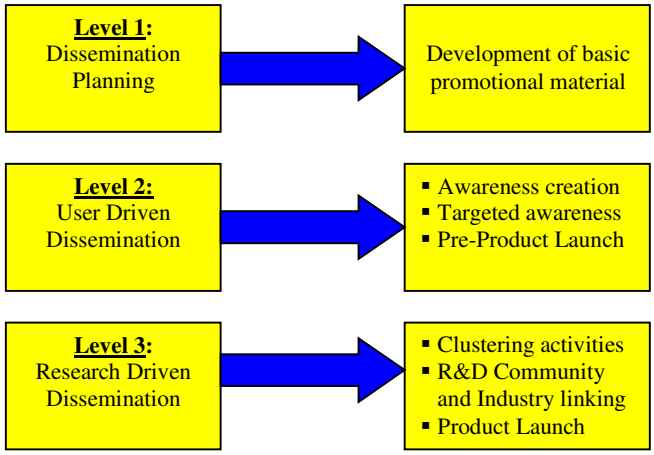


Figure 3.1: The three levels of ONBASS dissemination

3.5 ONBASS Dissemination Activities

Initially, the ONBASS Consortium designed, prepared and deployed a series of means & tools ('dissemination material') to be used so as to maximise the impact of the project dissemination activities. This dissemination material encompassed:

- The ONBASS Logo: The project logo (see figure below) was eventually chosen out of a total of 9 candidate designs.

Figure 3.2: ONBASS Logo

- The ONBASS Project Presentation: The project presentation was developed soon after the kick-off of the ONBASS project in order to accommodate any potential contacts made by the partners or the consortium and while the ONBASS Project Brochure was still not finalised.
- The ONBASS Brochure: The ONBASS Project Brochure aims at informing an individual who has no prior knowledge of the project, on the project's aims and objectives. Using an eye-catching and appealing graphics background and through the selected texts, photos and schemas, the major objectives, scope, innovation and benefits of ONBASS are presented.
- The ONBASS Web-site: The ONBASS Project Website can be found at www.onbass.org. The main objective of the website was to provide information about the project to the public as well as to serve as a collaboration mechanism for the consortium members. The website consists of a public and a restricted members-only access area.

- The ONBASS Newsletter(s): In total, five (5) newsletters overall were produced as part of the ONBASS project. These newsletters were produced every six months after the end of the 1st year of the project. The purpose of the newsletter(s) was to create targeted awareness to a number of industry key players and actors building on some of the business contacts of the ONBASS consortium members and the contacts made during the course of the project. In following, a screenshot of one of the newsletters produced so far is provided:

Figure 3.3: ONBASS Newsletter Screenshot

Further, all ONBASS partners ensured that a reference to the ONBASS project was made on their corporate website while also a link was established between their websites and that of the ONBASS project itself. This way all individuals or organisations coming into contact with the partners of ONBASS could be informed of the project, its objectives and results.

Thereafter a series of press-releases and publications were issued by the ONBASS consortium targeted at creating awareness in the related industry regarding the project. The most prominent of these 'activities' included:

- An interview regarding ONBASS conducted with the renowned 'Flight International' magazine (see below):

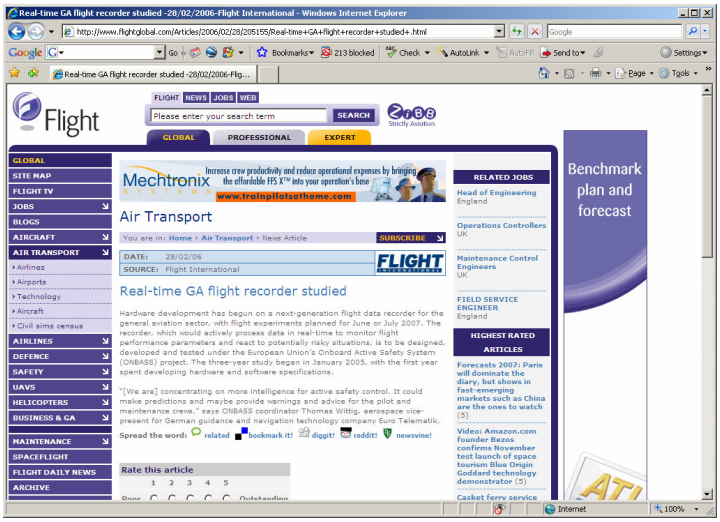


Figure 3.4: ONBASS story on 'Flight International'

- A press-release published by the "ASD Network" (Global Aerospace & Defence Network) portal. This is a major news network of the Aerospace Industry which circulates headlines of related industry stories to its members. It members include key industry players. The press-release prepared was circulated to the subscribers of the site on the 06/12/2006 (was one of the top stories of the portal on that day) and is still accessible today on the 'ASD-Network' portal (http://www.asd-network.com/press_detail_B.asp?ID=6096). A screenshot of this press-release follows below:

Figure 3.5: ONBASS Press-Release on 'ASD-Network' Portal

- A press-release published by the "Remove Before Flight" bi-monthly, aviation news, stories & features magazine in the January-February issue (11/01/2006), which can be viewed below:

Figure 3.6: ONBASS Press-Release in the 'Remove Before Flight' Magazine

Additionally, the ONBASS Consortium attended and/or presented and/or held discussions with interested parties at a series of industry events, conferences, workshops and seminars, some of the most prominent such 'cases' being:

- The 'Swiss Aeroday 2006', where LondonMet attended, made contacts and handed-out ONBASS brochures to a number of the attendees.
- The '2nd European Conference for Aerospace Sciences' (EUCASS) - in Brussels in July of 2007 - where 2 papers were submitted and presented by ONBASS consortium representatives under Symposium 3: 'Avionics, Flight Dynamics & GNC'.
- The MAKS [astec' 07, "New Challenges in Aeronautics"] conference (Moscow, August 19–23, 2007) where ONBASS consortium representatives attended and presented a paper, held discussions regarding the project context and handed out leaflets to representatives of major aerospace industry bodies and organisations.
- EUROCONTROL's 6th INO Workshop (4-6/12/2007, Bretigny), where ONBASS consortium representatives submitted a paper (which was accepted for the 'Poster Section' of the workshop), attended and supported the presence of the 'ONBASS Poster', carrying-out very fruitful discussions with regards to the project, its results and future collaborations with interested parties. A considerable number of ONBASS Brochures were also handed-out as part of this Workshop. Screenshots of the paper, poster and workshop follow below:

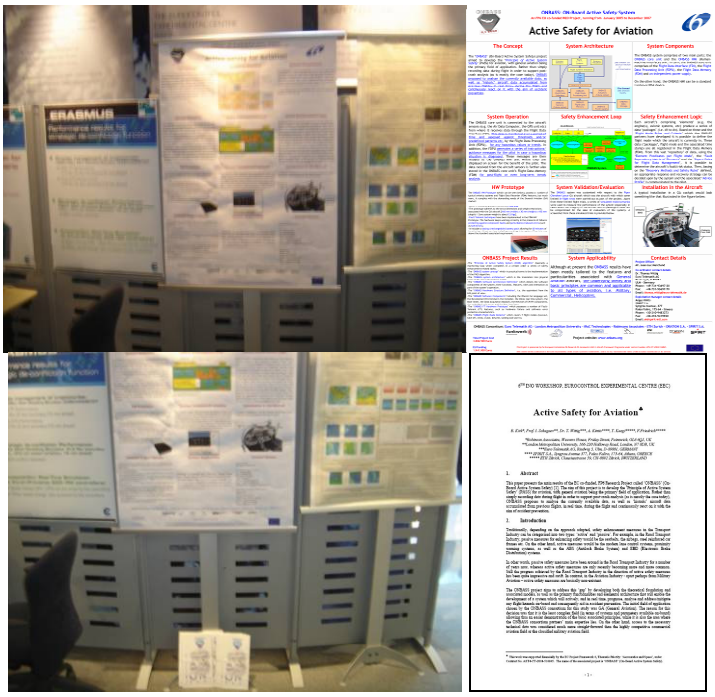


Figure 3.7: Screenshots related to the ONBASS Poster at the 6th INO Workshop

4 References

[1] Annex 1 - "Description of Work", Version 1.1, dd. 12/11/2004, for ONBASS. Contract of the 6th European Framework Program: "*Integrating and strengthening the European Research Area,* Thematic Priority: Aeronautics and Space, Contract No.: AST4-CT-2004-516045.
[2] Deliverable D0.1 Project Management Plan, ONBASS Project
[3] Deliverable D1.1 Application Domain Definition, ONBASS Project
[4] Deliverable D1.2 Reliability Model Description, ONBASS Project
[5] Deliverable D2.1 System Concept and Structure Definition, ONBASS Project
[6] Deliverable D2.2 System and Application Specification, ONBASS Project
[7] Deliverable D2.3 Verification and Validation Plan, ONBASS Project
[8] Deliverable D3.1 Software Architecture Definition, ONBASS Project
[9] Deliverable D3.2 Verified Software Modules, ONBASS Project
[10] Deliverable D4.1 Hardware Structure Definition, ONBASS Project
[11] Deliverable D4.2 Verified Hardware Prototype, ONBASS Project
[12] Deliverable D5.1 Integrated ONBASS Demonstrator, ONBASS Project
[13] Deliverable D5.2 Verified ONBASS Demonstrator, ONBASS Project
[14] Deliverable D5.3 System Demonstration Report, ONBASS Project
[15] Deliverable D5.4 System Evaluation Report, ONBASS Project
[16] Deliverable D6.1 Project Brochure, ONBASS Project
[17] Deliverable D6.2 Knowledge Dissemination Plan, ONBASS Project
[18] Deliverable D6.3 Project Profile, ONBASS Project
[19] Nordian Principles of Flight, Second Edition, ISBN 82-8107-054-4
[20] Piper Cherokee Lance Instruction Manual, PA-32R-300, Part Number 761-633
[21] Test Automation Framework presentation from Oberon Day
http://www.oberon-industry.ethz.ch/event/presentations/p_kirk

5 Abbreviations

a/c	Aircraft
ADC	Air Data Computer
BCD	Binary Coded Decimal
BIT	Built In Test
COTS	Commercial Off The Shelf
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
EMC	Electro Magnetic Compatibility
EN	European Norm (standard)
FDM	Flight Data Memory
FPGA	Field Programmable Gate Array
GPS	Global Positioning System
HMI	Human Machine Interface
ICAO	International Civil Aviation Organisation
JTAG	Joint Test Action Group http://en.wikipedia.org/wiki/JTAG
KIAS	Knots Indicated Air Speed
LED	Light Emitting Diode
MFS	Microsoft Flight Simulator
MMC	Multimedia Memory Card
ONBASS	Onboard Active Safety System
PASS	Principle of Active Safety System
PDA	Personal Digital Assistant
SLIP	Serial Link Internet Protocol
TBD	To Be Determined
UART	Universal Asynchronous Receiver Transmitter
WP	Work-Package
XML	eXtensible Markup Language