ICAS-92-2.3.1

HOW TO MASTER THE INCREASING COMPLEXITY OF SYSTEMS ON BOARD CIVIL AIRCRAFT

Y. Nègre - Head of Systems Department
Aérospatiale - A/DET/SY
316 Route de Bayonne
31060 TOULOUSE

Abstract

After showing how the complexity of digital systems has doubled about every five years, and evoking the difficulties caused by this evolution, a methodological approach is described called the "Systems Development Workshop" aiming to gain mastery of this evolution.

Among the different phases of the system : design, production and validation process, the importance of the design phase is emphasized. This design activity is analysed in terms of tasks to be performed and means to be implemented. Particular stress is laid on the power and benefits that can be expected from computer design tools.

Taking into account the multitude and diversity of means to be implemented, the importance of ensuring consistency between these means is shown. A dedicated host environment is required for management, communication, documentation, and convivial support of system developers.

Finally, it is noted that the increasing necessity of international cooperation in civil aviation consolidates the proposed approach.

I - ONBOARD SYSTEMS : INCREASING COMPLEXITY

The complexity of digital systems on board civil aircraft is constantly increasing.

Figure 1 shows that for the AIRBUS programs, this complexity has been multiplied by a factor of 2 between each program : A310, A320 and A340, in other words, every 5 years.

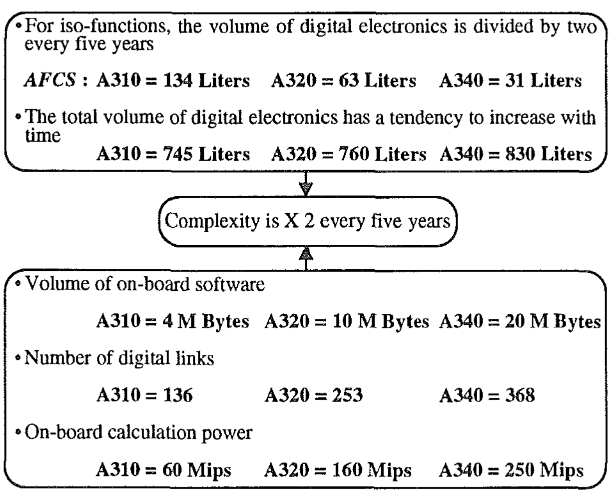


Figure 1 - Evolution of digital systems

Simultaneously, a constant reduction in the volume of electronics required to perform a given function has made it possible to keep the volume of the onboard electronics more or less the same, mostly thanks to VLSI (Very Large Scale Integration) technologies.

Examining the reasons for this rapid growth in the complexity of the systems, one can be sure that this will continue for the next 10 years. The reasons for this are :

- The increasing computerization of the systems. Concerning this point, development is tending towards higher "multiplexing" of the functions ; in concrete terms, this is reflected by the arrival of integrated modular avionics and multiplexed buses of the ARINC 629 type ;
- The implementation of new systems. Yesterday, it was the fly-by-wire controls and the electronic instrument panels ; tomorrow, the structural flexibility phenomena of wide body aircraft will require active control to improve passenger comfort. In addition, one must foresee the development of navigation and communications systems incorporating such new functions as GPS (Global Positioning System) or greatly enhanced ground/aircraft dialog through digital links : ATM (Air Traffic Management) and ADS (Automatic Dependent Surveillance).
- The ever increasing interconnection between systems which inflates the volume of information exchanged. This is a natural development mainly linked to the automation of secondary tasks, allowing the pilot to concentrate on flight control.

Lastly, note that onboard systems directly influence flight safety, more and more, thus requiring tight controls for design and manufacture.

II - CAN THE DEVELOPMENT OF COMPLEX SYSTEMS BE MASTERED

The more a system is complex, the more difficult it is to guarantee the date on which it will be set into service and to keep costs within the initial budget. Experience in this field, and not only in aeronautics, shows that debugging systems such as these requires a lot of modifications and that the later these modifications are embodied, the higher the costs incurred and the time required.

One example, well known by the aeronautical community, is the Flight Management System (FMS), the debugging of which required several thousand modifications, with delays of several years with respect to the planned certification dates.

In addition, over and above the intrinsic complexity of certain systems, note that systems which have a high interaction with man are by nature difficult to debug. Only permanent dialog with the future users (crews, etc...) from the pre-project phase will allow a satisfactory definition of these systems to be obtained.

Our aim is therefore clear, we must :

- minimize the number of modifications,
- discover them and embody them at the earliest possible stage.

The purpose of the following overview is to propose a rational and methodological approach for the development of systems within a structure called the Systems development Workshop.

This approach must allow the objectives set above to be achieved.

III - SYSTEMS DEVELOPMENT WORKSHOP

Before explaining in more detail what a systems development workshop is, we shall specify the domain covered by this workshop.

1 - Domain covered :

We can determine this by listing the various chronological phases required to deliver a "ready to go" system.

The conventional presentation in the form of a V is given on Figure 2. There are three separate phases : design, manufacture and validation.

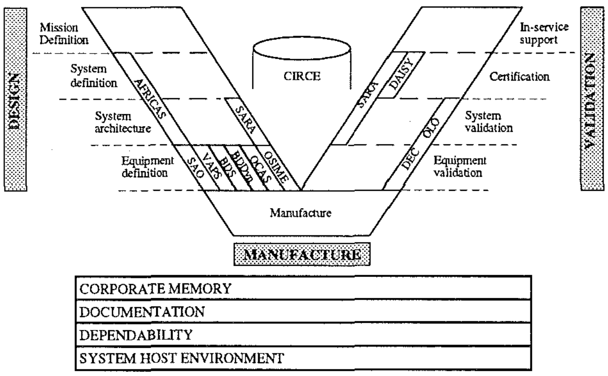


Figure 2 - System development cycle and associated tasks

In the V presentation, a validation step corresponds to each design step. Thus presented, the domain of the system workshop is vast as it covers the three above mentioned phases.

However, note that the manufacture phase is, in general, entrusted to the equipment manufacturers apart from the software section of certain equipment considered as critical. This intermediate phase is part of the system workshop visibility domain as one must be capable of following the progress of the manufacturing activities and ensuring technical traceability.

2 - Definition :

"The systems workshop is the coherent set of methods and tools required for optimum development of systems within a given aircraft program context".

3 - Description :

The best way to describe a Systems workshop is to review the various phases of the process (*DESIGN*, *MANUFACTURE*, *VALIDATION*) to examine, for each of these phases, the corresponding requirements and the solutions that can be found.

Special emphasis is placed on the DESIGN phase as this phase is critical for the final product quality level ; within this phase, it will not be possible to examine all the steps and therefore all the tasks comprising each of these steps : only the most significant ones will be considered.

Also, a certain number of global tasks, active in all the phases of the process, will be described.

3-1 - Design phase

• System architecture design

Upstream of this phase, all the functions to be fulfilled must be defined and the criticality levels of each of these functions must be specified. From this point on, the system architecture design can be undertaken considering :

- the constraints specific to the aircraft,
- available technologies;
- the natural grouping of functions,
- the cost, weight, overall dimension and maintenance objectives, etc...

Re-use of the system architecture of previous aircraft as a starting base for the new one is a common and cost-effective practice in civil aeronautics. This method cannot apply, though, when developing a brand new system, or when using emerging technologies. As an example, Integrated Modular Avionics (I.M.A.) can call for a complete redesign of system architecture. In this case, the possibility of using common resources for several functions requires that the distribution of the hardware resources and the distribution of the functions in this hardware be tackled together. A tool dedicated to designing the architecture of these types of system is being experimented at Aerospatiale.

Its name : AFRICAS (Analyse Fonctionnelle et Répartition Interactive pour la Conception d'Architectures Systèmes : functional analysis and interactive allocation for system architecture design) is representative of the aim pursued.

Figure 3 shows the design loop followed. Generally, starting from a basic hardware design we make a functional breakdown and for each function a redundancy choice to meet the criticality level of the function concerned, then we allocate the functions and their redundancies to the hardware. We then assess this first architecture by comparing it against the segregation, consumption, availability, safety requirements, etc... Then, if necessary, this is repeated one or more times.

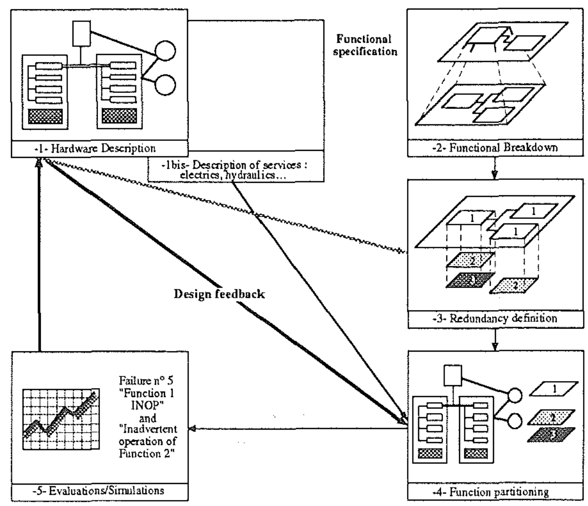


Figure 3 - AFRICAS : Functional analysis and interactive partition for system architecture design

Note that integrated modular avionics is the most complex case to deal with from the architectural design point of view, as there are :

- several nested levels : System, Equipment, Module, etc...
- links (data flow) between the functions to be taken into account.
- a major impact on the traditional industrial work sharing.

This task results in a specification of the system which includes, among other things :

- the functional requirements,
- the safety requirements,
- the interface with the other systems,
- the physical characteristics,
- etc...

• Equipment specification :

An item of equipment can be defined from two types of specifications, either a general specification which lists the functions and performance to be ensured, or a detailed specification which describes the execution algorithms of these same functions.

Aerospatiale has chosen to specify in detail the items of equipment intimately linked with the operation of the aircraft, that is mainly, the critical equipment :

- fly-by-wire computers, autopilot, man/machine interface, etc...

Originally, the specifications were written in natural language ; the overrichness of this type of language led to software coding errors, as the specifications could be incorrectly interpreted by the programmer analyst. With this in mind, Aerospatiale developed a methodology based on a tool designated SAO (Spécifications Assistées par Ordinateur : Computer-Aided Specification). The major originality of this tool is its graphics language which uses a symbol library and assembly rules known by all electronics and automation engineers. This language covers the field of operational logics and closed-loop systems ; an example of a sheet with SAO formalism is shown on Figure 4.

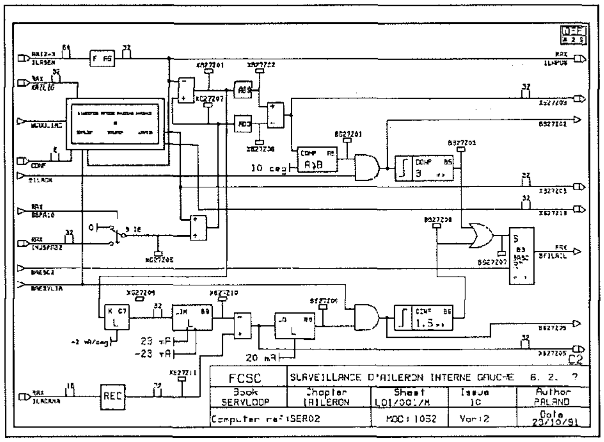


Figure 4 - SAO : computer aided specification

This formalism allows :

- the specified function to be readily understood without ambiguities and therefore avoids most coding errors,
- changes to be controlled and therefore the traceability of the specification to be ensured,
- a consistency check to be conducted on each sheet, then on the specification.

Lastly, SAO allows automatic coding. This is not the least of its merits, as automatic coding has the following advantages :

- reduction in coding errors to a level which is practically null,
- elimination of unit tests,
- reduction in the software manufacturing cycle, especially the modification cycle,
- possibility of validating the functions of an item of equipment as early as the design stage. This point is covered by the following paragraph.

The VAPS tool (Virtual Anything Prototyping Systems) made by the Canadian company VPI (Virtual Prototype Incorporation), associated with the SAO tool allows cockpit display symbologies to be defined with prototyping and animation of symbols.

This task results in a specification for the equipment concerned. It comprises :

- a part specified under SAO and/or VAPS :

  • flight control laws and/or operational logics,
  • equipment input/output signals,
  • display symbologies.

- a more conventional part concerning, among other things :

  • safety requirements,
  • physical characteristics,
  • environmental constraints,
  • etc...

• Validation of specifications :

First of all, for reasons of clarity, we shall distinguish between validation and verification.

The verification of a product consists in ensuring that this product is in compliance with the specification defining it. The validation operation consists in ensuring that the product specifications are correct and complete. The importance of this operation can be easily seen as it would be pointless to know how to code a specification automatically, without making errors, if the specification itself contained errors or was incomplete.

As we have already seen, this validation operation is covered by the bottom-up part of the V and we will discuss later the means implemented to achieve it, such as ground tests on a simulator, and flight tests. The advantage of having means to validate the specifications as soon as they are written without having to wait for the manufacture of the equipment is obvious.

The means installed at Aerospatiale for the designers take the form of "in-house" simulators. To illustrate the attractiveness of this type of tool, take for example the fly-by-wire and autopilot control law specification :

From the aircraft control targets, defined in terms of rigid aircraft mode damping and uncoupling, we design the aircraft control laws using automatic multivariable analysis methods.

The laws obtained in transfer function form are then explained in SAO formalism observing a top-down breakdown : book, chapter, sheet which allows the specification to be structured.

These sheets are then coded automatically and the code obtained loaded into a real time computer which simulates the aircraft flight mechanics. A control panel, including the mini-flight controls and a display showing the main flight parameters, make it possible to "fly" and validate "hands on" the resultant control software - within minutes ! If one of the design parameters is not satisfactorily met, we just reiterate.

Figure 5 gives a representation of this mini-simulator known by the name of OCAS (Outil de Conception Assistée par Simulation : Simulation-assisted design tool).

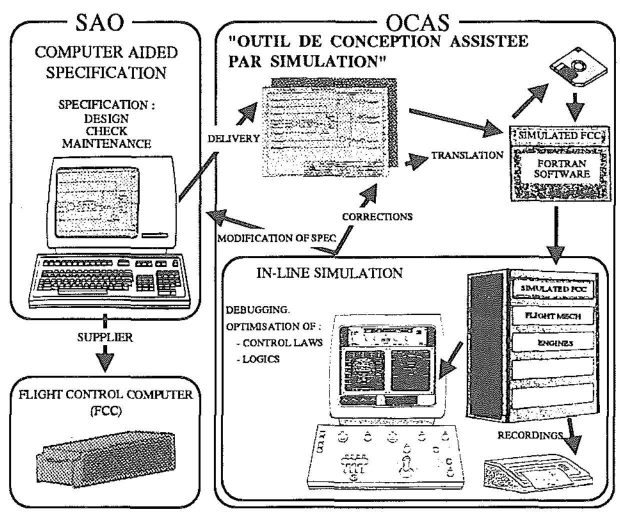


Figure 5 - OCAS : simulation assisted design tool

This tool therefore allows the aircraft control laws to be validated at an early stage in the development cycle. The check made is a functional check which does not take the hardware aspects at the monitoring and redundancy levels into account.

These aspects are taken into account by another tool which can simultaneously simulate the operation of five computers, each computer including a "Control" channel and a "Monitoring" channel. We can therefore validate :

- failure-free operation at tolerance limits,
- operation with failures and downgraded modes,
- cross-computer synchronizations,
- transient effects,
- feedback loops with simulated servocontrols.

This tool, which is extremely powerful, allows several hundred cases to be dealt with overnight.

Here again, automatic coding of the SAO sheets allows the modification cycle to be reduced to a few hours.

This tool is known by the name of OSIME (Outil de SImulation Multi-Equipements : Multi-equipment simulation tool). A representation is given on Figure 6.

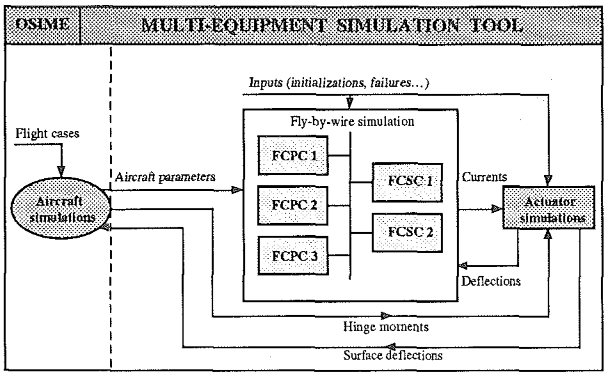


Figure 6 - OSIME

• Interfaces between systems :

This is one of the most difficult problems to master, because, firstly, of the very large number of data and information exchanges which exist and, secondly, because of the asynchronous operation of the various computers.

To address the first problem, Aerospatiale has made up a data bank of signals exchanged in the aircraft in order to obtain centralized management. The resulting functions are :

- description of the signals exchanged,
- management of interfaces between equipment,
- consistency check on these exchanges.

To give an idea of the type of data and the exchanges, Figure 7 shows a theoretical diagram of the dependence of these data.

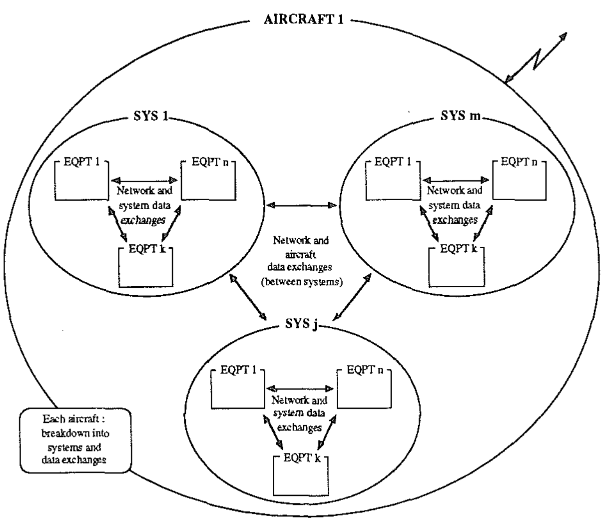


Figure 7 - Aircraft data exchanges

The second problem concerns the "dynamic" interface of the electric signals exchanged. The asynchronous operation of onboard computers can generate temporary disagreements, which are often troublesome for the crews, especially in the case of electrical transients and on automatic reinitialization of one of the computers. The large number of equipment manufacturers, each with their own interpretation of the exchange rules (ARINC 429 standard) only increases the difficulty. Today, this problem is dealt with by a consistency check on all the signals exchanged between computers.

By addressing this problem from the design stage, we know how to reduce the number of spurious indications and/or unwanted disconnections of the onboard systems.

• Definition and installation of electrical wiring :

The length of the electrical wiring installed on an aircraft totals 180 km ; production of the electrical drawing set and its management represents a considerable task ; the same applies to the routing of these cables which must comply with stringent segregation and installation rules. To deal with this task, we have developed a tool called CIRCE (Conception Informatisée et Rationalisée des Câblages Electriques : Rationalized and computerized design of electric cables) which allows :

- wiring diagrams to be produced in CAD,
- the list of cables to be manufactured to be extracted in computer file form,
- these cables to be allocated to the assemblies (harnesses) and to their subassemblies (which correspond to manufacturing operations),
- data bank management of :
  • standard items,
  • diagrams,
  • cables,
  • cable terminations,
- the routing of the electric cables to be defined in 3D CAD.

The latter function, associated with automatic calculation of bundle lengths allows the requirements of the Production, Inspection and Product Support departments to be met from the design stage.

3-2 - <u>Manufacture stage</u>

In Aerospatiale's industrial organization, manufacture is ensured by the equipment manufacturers, the System Workshop only has a visibility and work follow-up objective for this phase.

We make however an important exception for the software part of certain items of equipment, for example the flight controls. These are the application software packages that we develop ourselves within a software production workshop. The main characteristics of this workshop are given below.

• Software workshop

The development of the flight control software packages calls to a large extent on specific methods and tools. In fact, we tried to get the best out of the way the specifications were produced by the system workshop facilities.

An automatic code generator allows 70 % of the software of the main computer to be obtained from the SAO specifications. This tool had to be developed with the same degree of quality as embedded software.

AEROSPATIALE has also designed a test sequence-assisted generation tool. This tool is used to check that all the system functions specified in SAO language have effectively been coded.

Lastly, a software configuration management tool identifies changes in the specifications and automatically determines the operations required for updating the software accordingly.

Thus, by simplifying the manual tasks in the complete production cycle and by close coupling between the management of the "systems" specifications and the management of the software packages, AEROSPATIALE can comply with the level of quality required for its critical systems.

3-3 - Validation phase

We have already touched on this subject, the system validation considerations must be borne in mind, as early as possible, by the designers. We have already described some of the means allowing this to be achieved ; from here, we must continue the validation on the real equipment. We will not describe the product verification tests, as these tests are conducted by the equipment manufacturer on his premises according to procedures and documents which must be approved by the aircraft manufacturer.

• Ground tests

Distinction is made between :

- tests on partial benches which allow the operation of each of the systems taken separately to be tested with simulation of the peripheral systems,

- tests on the integration bench which allow the cross-operation of the systems to be tested. We try to make this integration bench representative of the aircraft as regards the geometrical shape, the cables and the power resources such as the electrical and hydraulic power distribution and generation systems.

- tests on the simulator conducted to validate the aircraft control laws and the failure procedures in a real environment. This simulator is therefore equipped with a cockpit similar to the one on the aircraft with a simulated view of the outside world. Aerospatiale's experience on this subject shows that it is not necessary to move the cockpit to simulate the aircraft movements. This simulator therefore differs in this respect from the training simulators used in the pilot training centers.

These tests on simulators with pilots are the natural extension of those conducted on the "in-house" simulator called OCAS. In both simulation cases, the identity of the assessed software packages is ensured thanks to the use of SAO and automatic code generation.

• Flight tests

This is the ultimate phase. In an aircraft development cycle, this phase lasts approximately one year whereas the previous phases are, in general, spread over several years. The tests on the systems represent only a part of the 1200 flight hours required for the complete certification of the aircraft ; the debugging of the aircraft is carried out, on the one hand, with the flight test crew and, on the other, from the information recorded during the flight tests. For this, the aircraft must be instrumented with high-performance recording systems in order to be sure that all troublesome transient phenomena are captured. Several thousand parameters are thus recorded during each flight, either analogically for large passband signals or digitally for the others ; the recorded data flow represents 200,000 bits per second of flight. This information is recorded onboard the aircraft but part of it is transmitted directly by telemetry to the ground allowing the test data to be processed in real time.

Searching for the causes of anomalies which lead to disconnection of systems or unwanted warnings, is a real brain-teaser as, on account of the asynchronous operation of the computers, we cannot simply recover the chronology of the various logic signals which makes it very difficult to distinguish between the causes and the effects.

Aerospatiale has developed an analysis tool called DECOLO (DECOnnexion LOgique : Logic disconnection) which automa-tically diagnoses logic-type anomalies.

During a test flight, certain anomalies are thus explained before the aircraft lands. The principle is as follows :

From an event which triggers a change in Boolean state, the analysis program executes the following operations : it first identifies the variables upstream of the event concerned. On the SAO sheet comprising these variables, we simulate the input variables with the value read on the flight recordings and compute all equipotential lines. Then, by simply using a backward analysis on the SAO sheet we can deduce the incriminated input or inputs. Figure 8 illustrates this principle, again made possible by the fact that the specifications are written with the SAO formalism.

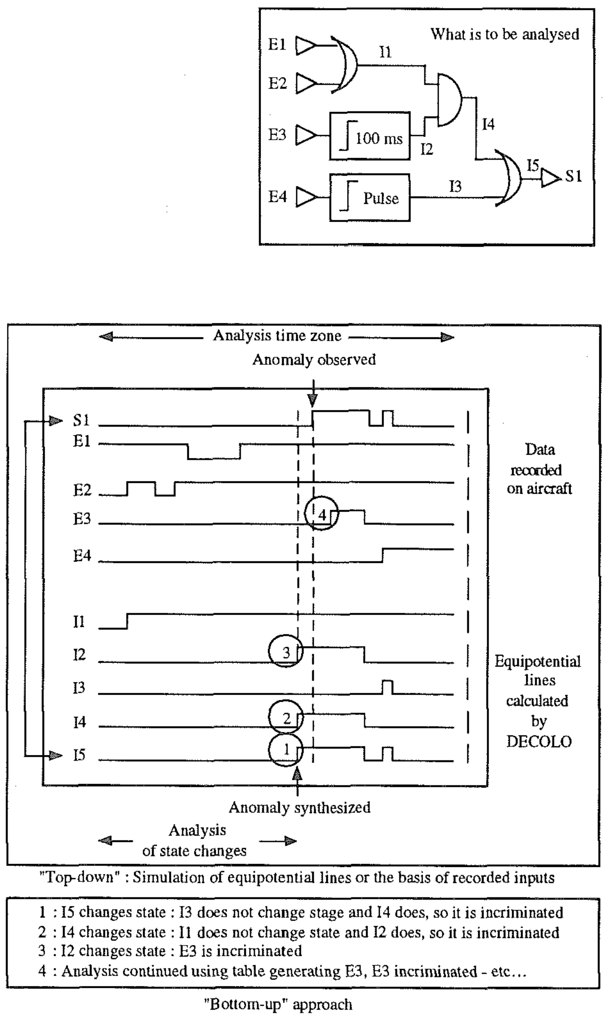
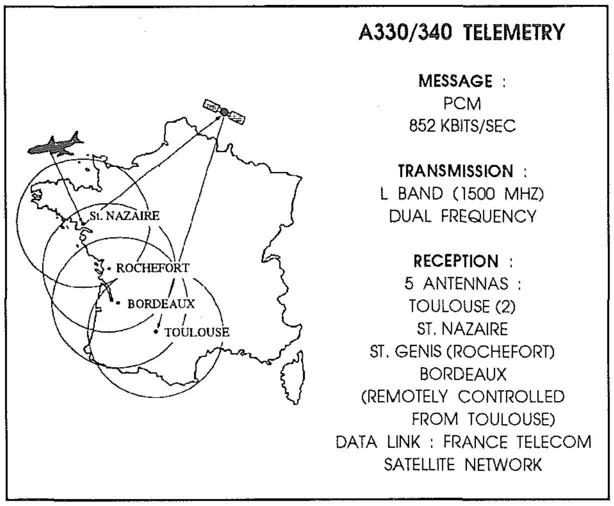


Figure 8 - DECOLO : logic disconnection
Principles of table analysis

The tool that we have just described is just one item of the very important set of methods and means which must be brought into play to settle, during this flight test phase, the last problems which remain in spite of the efforts made in the previous phases. Figure 9 shows, in a simplified manner, the means developed for telemetry.



3-4 - Global tasks active throughout all phases of the process

Some tasks are active during the whole process.

Figure 2 identifies four of these tasks :

- Corporate memory :

Past experience or the company's memory must exist in an easily-accessible and user-friendly form, this is rarely achieved.

From the information required by the designer, we must single out :

- The technical directives and the regulatory texts : the pile of documents covering all the systems of an aircraft stands more than 6 feet high ! In fact, the designer needs a guide to find the information that he requires in all this undergrowth.
- The selective experience which results from actual debugging cases or even incidents or accidents. To perpetuate this experience, Aerospatiale has transcribed these results into sheet form and has extracted the applicable "Rules", with a procedure for following up the applications.
- The know-how or, again, the expertise which makes up the company's culture and ways of reasoning ; this concerns more the whys of earlier designs rather than the wherefores.

Aerospatiale is developing a program called MERE (Mise En Règle de l'Expérience : experience set into rules) which is a global corporate memory project taking into account the various above mentioned aspects.

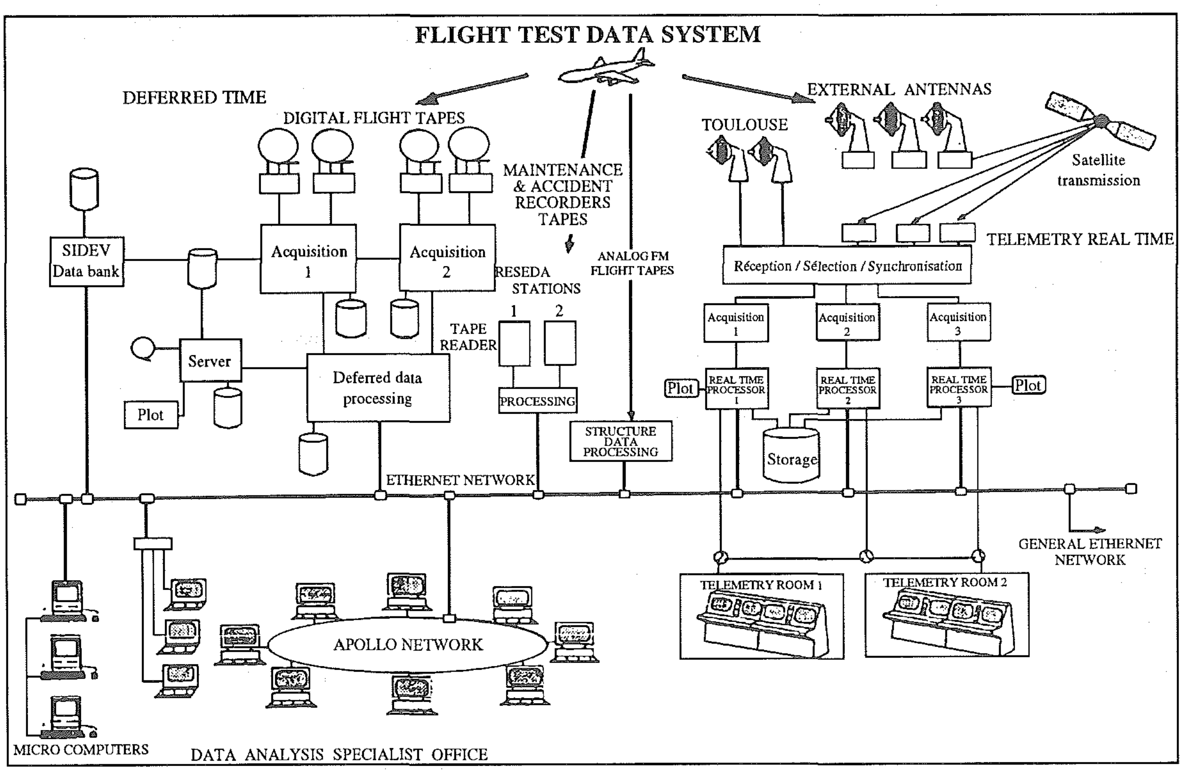


Figure 9 - Flight test data processing

Figure 10 explains the various components of this project. This perpetuation of the company's know-how seems to us to be fundamental ; it applies upstream of and throughout the complete design process.

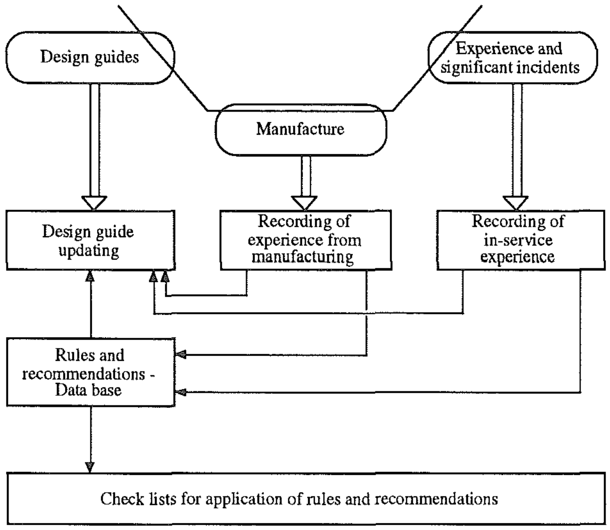


Figure 10 - Corporate memory organization :
MERE : experience set into rules

• Documentary chain :

Today, most of the documents produced are not always derived from one another, there is no systematic chaining between documents.

Lacking methodology and standards, the documentation of complex systems soon becomes difficult to control.

To produce, manage and use the numerous documents which result from the design, manufacture and validation phases of a system, we need :

- a documentary tree-structure model. This model can be built by analyzing the designer's tasks and the type of documents produced,
- a document writing guide,
- document chaining possibility,
- document configuration management,
- traceability for each of the documents.

As part of the documentation, drawn up during design, is used later for maintenance purposes, it seems to be a good idea to consider extending the SGML (Standard Generalized Mark up Language), already used for the Aerospatiale Product Support departments and other aircraft manufacturers, to the design phase. SGML is a standardized method which allows a document to be structured independently of the means used for its acquisition or its processing and independently of its physical form.

This standard facilitates exchanges. It has also been adopted by the DoD (Department of Defence) in the CALS (Computer Aided Logistic Systems) program.

• Dependability
(safety, reliability, availability, maintainability)

This is a vast subject with which system designers are confronted.

Without attempting to summarize in a few lines all the tasks allowing this dependability to be obtained, we can mention the most significant aspects.

This is a discipline which reflects the will to control the technical, human or environmental hazards. It is based, not only on the experience acquired (regulations, standards), but also on the ability to prevent risks by predicting them.

This is a continuous process and an integral part of the design, manufacture, validation and in-service operation phases.

Dependability

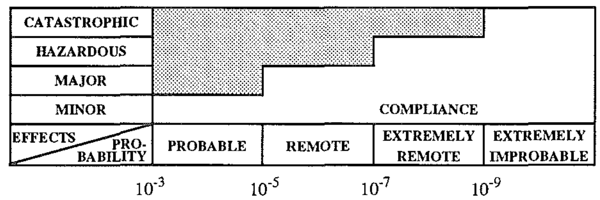


Figure 11 - Domain of compliance of events :
consequences versus probability

- In the system design phase : (Figure 11)

The prediction of functional failures and the assessment of their consequences allows :

- the criticalities of the functions to be defined,
- safety objectives to be allocated to the functional failures used as a basis for the architectural design of the systems : this procedure was used for the first time on Concorde.

- In the system installation design phase :

The prediction of external aggressions (lightning, icing) or specific hazards (engine burst, bird strike) makes it possible to define :

- the installation directives, segregation, for example,
- the design precautions, shielding of cables exposed to lightning, for example,
- the circuit protections, guards protecting the circuits against running fluids for example.

- In the manufacturing phase :

Dependability must integrate the quantitative and qualitative objectives from the system and installation design phases and meet them by :

- architecture design, redundancy, dissimilarity, etc...,
- quality assurance of the manufacturing process.
  For example, for onboard software where the stringent quality assurance measures, based on the RTCA DO 178 document, are applied.

- In the validation phase :

We must validate the product produced by demonstrating that the qualitative and quantitative objectives have been achieved.

Different methods are used in this demonstration :

• system safety analyses :

- top-down approach in accordance with the cause tree method.
- Bottom-up approach according to the elementary failure mode method : these modes allow each of the safety requirements to be followed up and therefore ensure their traceability up to certification of the aircraft.
  Two tools are available and widely used today within the scope of the Airbus, ATR and Hermes programs. The first tool named SARA (Safety And

Reliability Analysis) facilitates the work of the designer. It not only helps him in :

- data entry, issue and update of the document

but in addition, it provides :

- a consistent presentation,
- coherence within an analysis, between analyses and with other certified documents,
- demonstration justification management.

The second tool named DAISY (Dependability AIded SYnthesis) allows the analyses of each system to be linked, ensures their coherence and deduces the global safety level of the aircraft.

• analyses of specific hazards (for example engine burst).

- In the in-service support phase :

• dependability participates in maintaining the continued airworthiness of the aircraft,

• the airworthiness follow-up allows, with hindsight, the hypotheses made during the design phase to be confirmed.

• Dedicated system workshop host environment

There is little difference between the host environement of a software workshop and that of a systems workshop. In both cases, the aim is :

- to supply a user-friendly interface in order to guide the user when accessing the various tools and functionalities of the workshop.

- to manage the objects which are :

• proper to the workshop itself,
• or produced by the workshop tools.

- to create or manage the links between all these objects :

• either for the traceability of the activities,
• or to make up the documentation.

- to provide project management facilities with access rights,

- to communicate with other workshops (in particular, the Equipment Manufacturers' Workshops),

- lastly, to offer parameterizing possibilities at the level of the terminology, the language and the development follow-up procedures. All these aspects can vary from one company to another.

With all these services provided in a uniform manner by all the tools, we will be able to ensure the coherence of the systems workshop via this dedicated system host environment. We cannot overemphasize the importance and the need for this coherence as without it the designers' tools would flourish with the same logic as can be seen in the distribution of poppies along the roadsides in the springtime.

To develop this type of structure, we need widely-used standards and interfaces to be able to accommodate a large range of tools available on the market. For example, the PCTE (Portable Common Tool Environment) standard already used for software workshops.

This standardization is even more necessary as we are compelled to work in ever closer cooperation. This is the case for Aerospatiale, within the scope of the European civil aircraft programs such as Airbus and ATR.

Figure 12 shows a diagram of the dedicated system host environment that we are gradually installing at Aerospatiale.

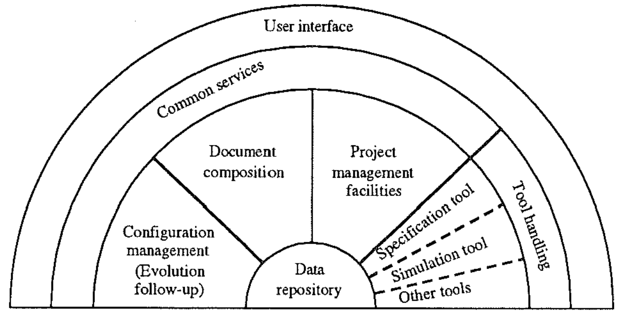


Figure 12 - Dedicated system host environment

IV - ENCOURAGING RESULTS

A systems workshop is being gradually set up at the Aerospatiale Aircraft Division. At the time of writing, still a lot remains to be done, in particular at the systems workshop host environment level. Also, the scope of SAO must be extended, on the one hand, by making it capable of performing new functions such as, for example, the addressing of data banks, certains types of scientific calculations, etc... and, on the other, by interfacing SAO with hierarchical functional analysis tools capable of complementing it upstream.

In spite of these needed improvements, the contribution made by some of these tools can be mesured.

SAO | COMPUTER AIDED SPECIFICATION

AIRCRAFT	A310	A320	A340
Number of digital units	77	102	115
Volume of onboard software in M.bytes	4	10	20
Encoding errors for 100 K.bytes software	A few hundreds	A few dozen	Less than ten

Note : On the A340, the computers benefitting from automatic encoding are :
• fly-by-wire controls - 70 %,
• automatic flight control - 70 %,
• display computer - 50 %,
• warning and maintenance computer - 40 %.

Figure 13 - Some figures

- SAO : Figure 13 shows the improvements achieved. On the A310, when the specifications were written in natural language, there were hundreds of coding errors on software packages with a size of 100 Kbytes ; on the A320, thanks to the use of SAO, this number was reduced to a few tens of errors and, today, on the A340, by adding automatic code generation, the number of errors is less than ten.

These results show that the generation of complex software packages, which was rightly considered difficult during the last ten years, is today a problem well under control at Aerospatiale.

- CIRCE : The contribution made by CIRCE affects two fields which are data quality and the development cycle :

    • the quality of the definition of electric cables has been considerably increased to reach the following percentage : 99 % of the cable definition is correct following the first data processing operation.

    • the processing cycle for a complete definition, has been reduced from 24 to 7 days.

  Note that a CIRCE operation today calculates the definition of 230,000 different electric cables, each one of them being described by 50 parameters.

- DECOLO : Automatic diagnosis of disconnections and/or warnings is an important step forward in the analysis of tests on Systems with complex operational logics.
  This tool provides good results in 80 % of the cases, in less than 3 minutes of analysis compared with the many hours of manual analysis required on previous aircraft.

V - CONCLUSION : REQUIREMENTS OF THE YEAR 2000

Our aim in this document was to make the reader aware of the benefits of having a systems workshop to design, produce and validate complex systems.

Much more than an exhaustive description of such a workshop, the aim of this document was to promote the approach leading up to it. Such an approach lives up to the aim pursued. Over and above the economical and lead time aspects, which alone justify the setting up of such a procedure, its fundamental strategic aspect must be underlined :

On the one hand, the communication and information transfer aspect, which is one of the goals of this project, is the key to the inter-partner cooperations required.

On the other hand, such a systems workshop is the cornerstone in the development of complex systems. With this asset, Aerospatiale is well on its way to mastering the complex systems of the year 2000 and beyond.