AVIONIC SYSTEMS FUNCTIONAL ANALYSIS AND SPECIFICATION

P. SCHIRLE
Avions Marcel Dassault - Bréguet Aviation
Saint-Cloud France

-:-:-:-:-

Abstract

The avionic systems for the 90's are characterized by their complexity, the limit of which we cannot see, and by the closer and closer integration of their various functions.

The improvement of the technologies during the past years has helped the digitalization and this trend will increase more and more : this digitalization implies the systematic use of data buses and the growing part of software in the equipment. With the advent of VHSIC, distributed processing, artificial intelligence, sensor data fusion, etc... the technologies are blurring the clear path of the operational and functional allocation defined for the "black boxes".

The avionic developments are now directed by a strict methodology, covering the definition, design and validation steps, especially in the software area.

From a quality assurance point of view, the most crucial phase of the development is the transition between the operational requirement and the functional specification of the corresponding avionic system. In other terms, it concerns the transition between "what the system should do" and "How the system is built".

The methodology used by AVIONS MARCEL DASSAULT puts a large emphasis on this phase, using original methods and specific associated tools.

After a brief reminder of the operational, technological and methodological context, the paper will describe first, the method and tools used, then all the lessons learned through their practical application.

It will point out :

- the requirement documentation set-up and verification,
- the functional analysis methods used, leading to the architecture definition,
- the functional specification and its verification through the use of rapid prototyping technics.

At last, a synthesis of the experiments done on previous programs such as RAFALE-A or MIRAGE 2000 NC, will be done and future trends will be mentionned.

1 - CONTEXT AND HISTORICAL ACCOUNT

The system activity is today a main branch of aeronautical industry. Its rise during the last two decades has been prodigious and modifies the aeronautical scene, creating new jobs and using new specific skills.

The evolution of the operational requirements has led to an increasing versatility which results in a tighter integration of functions in the system. The system itself is now shared between an airborne component cooperating with ground based components.

Originally restricted to traditional operational functions such as navigation, air to ground, bombing, air/air interception, systems have gradually grown richer, incorporating specific functions for sophisticated weapons, electronic warfare devices or reconnaissance equipement. The integration of these new functions aims at getting a maximum operational efficiency by :

- optimizing physical resources (sensors, actuators, data processing units) through data fusion and data exchange networks between aircraft and/or ground and maritime environments
- Optimizing human resources, through refined ergonomics of the man/machine interface, ensuring a high level dialogue with the pilots. The system selects useful data depending on the mission and presents them in a synthetic way.

Moreover systems are becoming highly evolutive. The operational envelope must evolve easily and the introduction of new functions or technological evolutions should not deeply modify the operation of the existing system.

From a technological point of view, avionic systems are characterized by :

- the use of new sensors implementing technics such as laser, infra-reds, CCD, etc...
- More and more compact electronics based on VLSI and ASIC circuits.
- More complex functional and physical architectures now integrate the new generation of sensors and historically independent systems such as engine, flight control system fuel or brake systems. This implies a generalized use of digital multiplexed connections between items of equipment.
- Massive use of software, bringing considerable flexibility but leading to specific problems of development control. Schematically, this evolution has led from decentralized systems, to fully distributed and integrated systems.

The present tendancy for the development of the systems is two-fold. Decentralization of processing power : the system software is distributed in a hierarchical way among several pieces of equipment.
Integration : each operational function is implemented through a series of functional modules spread over various item of equipment. The development then, involves a particular functional analysis, leading to a functional architecture distinct from the hardware architecture, thus blurring the clear operational allocation of each item of equipment.

The airborne avionic systems currently developped by "AVIONS MARCEL DASSAULT - BREGUET AVIATION" include more than a hundred items of equipment, half of which are mainly digital. Most of them are functionnally dependent on software. The volume of real time system software reaches several mega-bytes. The amount of data exchanged between equipment and/or functional modules exceeds 100.000 and the rate of delivery of data on the data bus is of several mega-bit per second.

The methodological revolution over the last years is the necessarily consequence of the operational and technological evolutions in order to keep the mastery of large systems development in terms of Quality and cost control.

Among the most typical factors, we can notice :

. the very important role of Quality assurance in design
. the modification of industrial organizations
. the specific problems of software
. the generalization of software tools supporting development

- Quality assurance in design

The importance and volume of the design activity relative to the development, the increased difficulty of evaluating the Quality (especially dependability) of the product delivered make the Quality assurance activities rise from the manufacturing level to the design level.

As the cost of modification, according to the moment when it is decided, increases in an exponential way during development, it justifies, that the maximum of energy must be spent at the very beginning steps of the design, that is to say during the requirement and specification phases.

The Quality of the product is demonstrated more and more through the evaluation of the development process, i.e. the qualification of methods used, than through the product itself. System Quality assurance and software quality assurance documents witness to this evolution. It has to be noticed that System Quality assurance includes software quality assurance and more traditional hardware production Quality assurance.

- Industrial organizations

The size of today's systems involves putting into common resources and skills distributed around numerous industrial companies (for instance, the number of people involved in the development of a MIRAGE 2000 avionic system exceeds 25000...). Strict methodology is the only way to support these new industrial organizations by enabling :

. the tasks and responsabilities identification and definition of each partner involved
. the assurance of harmonous development by reinforcing visibility and tracability

- Specific problem of software

The application software implemented within the equipment can be divided in equipment software and system software.

The purpose of the "equipment software" is to improve the separate performance of each item of equipment independently of its integration to the whole sytem. This kind of software may then been designed within a frame restricted to each item of equipement, by the equipment manufacturer himself.

The purpose of the "system software" is to improve not the individual performance of equipment by the global performance of the system, by ensuring, in a centralized way, the integration and allocation of the system resources in order to optimize the operational efficiency. This software represents an upstream functional layer as against the equipment software. It is also the result of a different design process : its definition results from the analysis of the entire system and not from the particular analysis of one item of equipment or of the function to which it is related.

Furthermore, specification of software consists, from a requirement expressed in term of "usefulness", in refining during successive stages and according to a repetitive process, the written expression of this requirement until it is given a shape which can be directly interpreted by a software machine : the code. The production of software only concerns its compilation and its reproduction. Design work of the software is of the same nature as design work of the system, the whole being therefore the fruit of a continuous methodological procedure.

Consequently, software development methodologies will be coherent with the system development methodology. Finally we can note that the specification work inherent to a software component of the system represents the sum of the specification work at every step of development (system then software), related to this component. This finding illustrates the non-obvious problems of ownership of the software !

- Software tools supporting the development

The methodologies of development, of the software first then of the system today are supported by more and more numerous and more and more sophisticated software tools. These tools are assistance to design, verification or validation and are grouped in workshops.

The appearance of the software has led to defining and setting into place software workshops which are nowadays numerous and varied but bearing very neighbouring methodologies in their principles. The appearance of the system approach, more recent, creates the need of system workshops, covering the software workshops and ensuring the assistance for the upstream design of these systems.

3 - SYSTEM APPROACH OF THE DESIGN

From the very beginning steps of the development, previous systems could be split into coherent and autonomous entities such as flight control system, engine control, etc... which could be designed separately. Complexity, integration and decentralization have modified the rules of the game.

System engineering, covering all the individual engineering for components design (sub-systems, items of equipment, software modules,...) is necessarily required to warrant performance and Quality. In fact, Quality and costs are mainly determined by system activities.

This system approach, to be efficient, requires the set-up of a system methodology whose aim is to define numerous steps of development, upstream of the equipment development steps. It allows to refine and to specialize progressively the design tasks, keeping in view the initial objectives of the global system.

It implies"wedding" designers of different skills, from the generalist in charge of the very first system break down in big functions, down to software and hardware specialized designers, going from a general and synthetic view of the system to progressively limited, detailed but coherent different ones.

4 - METHDOOOLOGY OF DEVELOPMENT, FORMAL TOOLS AND SIMULATION TOOLS

In this evolutive context, AMD-BA company has heavily invested to keep control of complex avionics systems :

- an original methodology of development of systems has been defined
- an appropriate System worshop has been designed and built
- Software workshops have been evaluated

The methodology precisely defines the steps of the development with related activities, products and means. There are design steps (descending branch of the "V" representing the life cycle) and symetric steps of validation (rising branche of the "V"), including the methodologies for software development are described.

The practical application of this system methodology leads to many problems in technical, organisational and human fields. Therefore, it involves a partitioning of design activities, leading to spread out the tasks among numerous designers of various skills. Furthermore, the different products relative to the design steps are merely documents, to be used down-stream in the cycle.

To get the maximum of efficiency from this methodology based on collective effort, but without demotivation of each individual actor, it is necessary to try to give each individual work a notion of "dlivered product" ; it means provide to each engineer, not only the information he needs, but also and above all the means to analyse, formalise and verify this own part of the work : this is the purpose of the system workshop kit of tools.

This workshop is made up of numerous software tools of two types :

- formal tools
- simulation tools

All the tools have common characteristics, required for their operation within the workshop : interconnectability, multi-user, multi-version, compatible digital environment. The different tools belong to a single environment structure which performs the following functions :

- project initialization
- development plan management
- system's configuration management
- integration of all specific tools (due to standard information exchange)
- access management

More particularly, it allows :

- the whole system definition and specification document set-up by composing the different products supplied by the various tools
- the global configuration management and the coherence assurance for all the elementary products, by controlling the modification mechanism of each tool of the workshop. This allows tool-aided completion of modification forms and the automatic up-date of the whole documentation.

Formal tools

The set of formal tools can be considered as an aid for a rigourous application of the methodology, through the improvement of communication, formalization of exchanges and management of the specification documentation.

Some tools are general purpose ones such as SCRIBE, a powerful documentation facility, or MITIA which supports graphic specification.

Others are given over to very specific technics and grouped in sub-worshops. Among them, the tools relative to dependability, grouped in the sub-workshop AXE, which allows safety analysis and evaluation.

Finally, most of the tools used are specific of a given step of the development cycle : their particular role will be described in the next chapter.

Simulation tools

The main purpose of simulation tools is to support the verification activities at each step of definition. They can also be used as design aids giving the engineer a living and quick feed-back of what he is doing, this allowing an interactive design/simulation/redesign approach.

Their main characteristic is a multi-area portability, allowing, through an effort of standardization to have on hand items of simulation which can be assembled as required, depending on the nature of each application. A description of each application will be given in the next chapter.

5 - STEPS OF DEVELOPMENT METHODOLOGY

5.1 - Preliminary definition

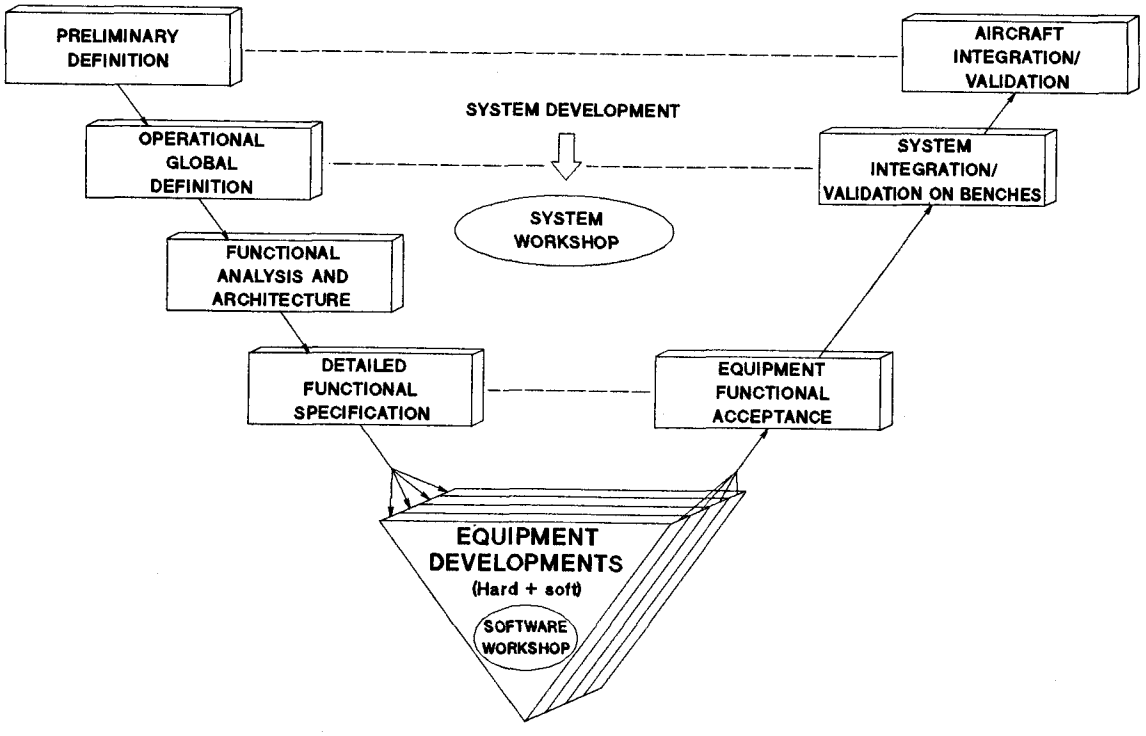
The role of this step is to define the general frame within which the system will be specified, taking into account an analysis of the requested missions and a first draft for the organization of the system.

The analysis of the missions relies on various feasibility studies and pre-studies of performance ensuring the global feasibility of the system terms of schedule and operational performance. It is concretized by a list of operational functions defining the mission envelope, concepts relative to the system's operating and a set of requirements for operating modes and performance of main sensors.

The study of the system organization leads to a first definition of the hardware architecture (equipment, stores, geographical fitting-out of the cockpit, fitting-out of the bays, etc...).

Formal tools

- general purpose tools
- C.A.D. tools as CATIA



Simulation tools

- The specific tool, SAMOS, allows us to illustrate, under a living form, the established concepts and then to help designers creativity, avoiding the "white page" syndrom, using a library of models (such as aircraft, sensors, or environment models). It permits the evaluation of operational scenarii. Each simulation item is defined and programmed by a mixed team designer/simulator.

5.2 - Operational global definition

The global definition step consists in defining precisely the objectives that the system must achieve (and not how the system will be built !).

The result of the step is the description, in terms of operational scenarii (therefore seen by the user) of the nominal operating mode of the system for all the functions it ensures. This description is made up of two types of documents :

- Guide-line documents
  These documents describe once and for all, the"general rules" applicable to every operational function with which the system is and will be fitted. They guarantee the coherent operation of the system. All these documents will be a reference for the set-up of the operational analysis of the system. (e.g. : guide lines for man-machine interface, failure display, maintenance, etc...).

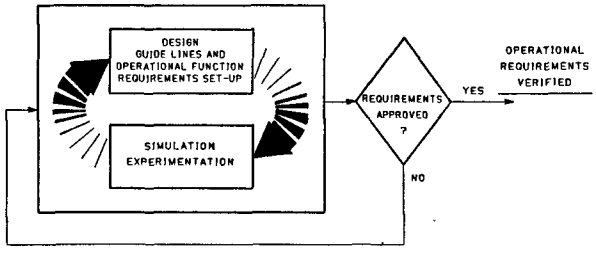
- Operational function requirements
  Each global specification document describes the operating scenario of the system for a given operational function. Each operational function is therefore specified in compliance with the guide-lines. Since these requirements, are independent, they can be elaborated autonomously and asynchronously. The coherence and the independence of the operational functions requirements, are a priori ensured by the guide-lines documents.

Formal tools

- General purpose tools

Simulation tools

- SAMOS tool
- OASIS tool : this tool is used for design and verification. Built around a representative mock-up of the cockpit, it supports a real time modelization of controls, displays and operational sequences, this enabling an efficient verification of the guides lines and operational requirements by the crew.



5.3 - Functional analysis and architecture

The role of this step is to analyse the system in order to build up the functional architecture. The functional architecture file (product of this step) describes a solution meeting the operational function requirements.

This step consists of two phases :

- construction of the functional architecture graph of the system
- integration of the functional architecture within the hardware architecture.

5.3.1 - Construction of the functional architecture graph

The method used consists of a progressive hierarchical breakdown of the system into functional elements, according to accurate criteria. Each successive level of breakdown involves :

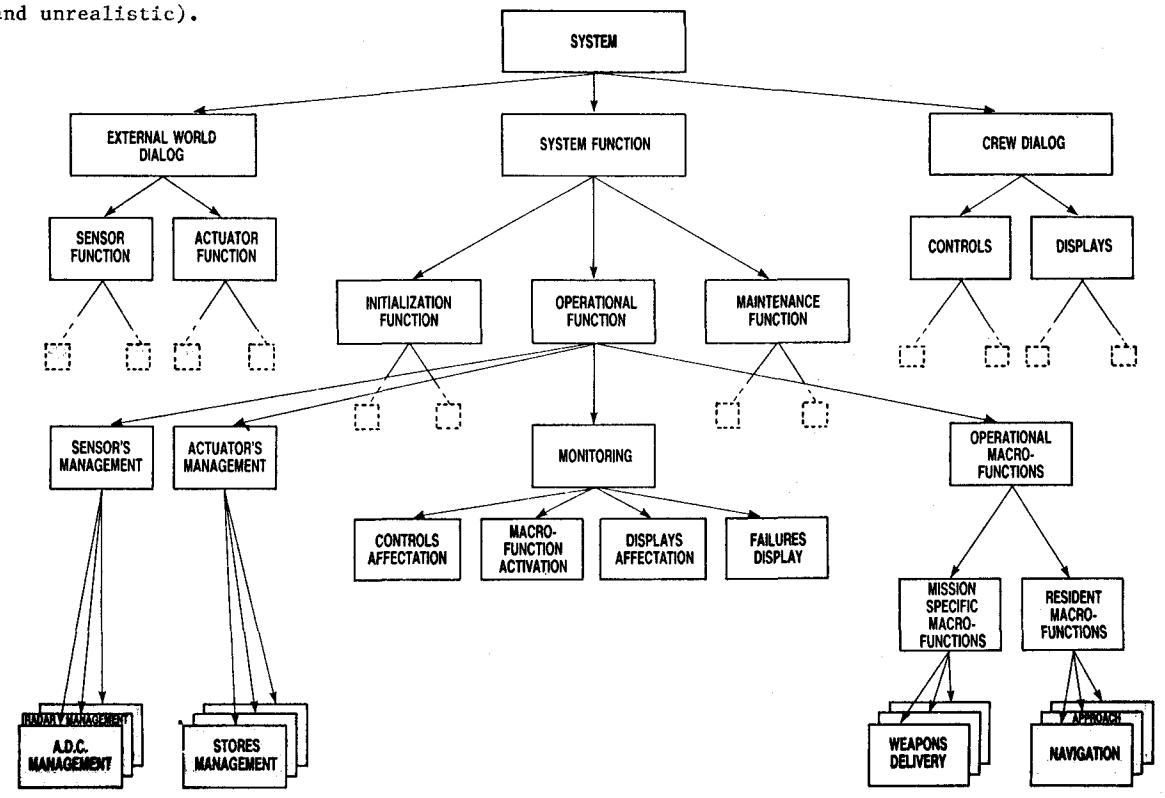
- a justification of the breakdown authorising understanding and approval of choices and constraints taken into account.
- a systematic collection of the interface between components of the architecture.
- a progressive refinement of the definition of these interfaces with respect to the breakdown level immediately above.

The functional graph describes the system according to a coherent tree structure. The group of the final components of the breakdown, so called functional modules, and their interface represent the functional architecture of the system. These modules will be specified in detail and then produced in hardware or software.

The constraints taken into account for the set up of the graph are :

- quality constraints and particularly flexibility ; these cosntraints impose independence and dependability rules between functional modules as well as rules for grouping tasks in the modules.
- operational constraints expressed in the guide-lines documents. Their analysis permits the definition of the "functional heart" of the system. This group of particular modules ensures the management of the other modules of the architecture.

The following figure gives an example of a system functional graph (through incomplete and unrealistic).



5.3.2 - Integration of the functional architecture within the hardware architecture

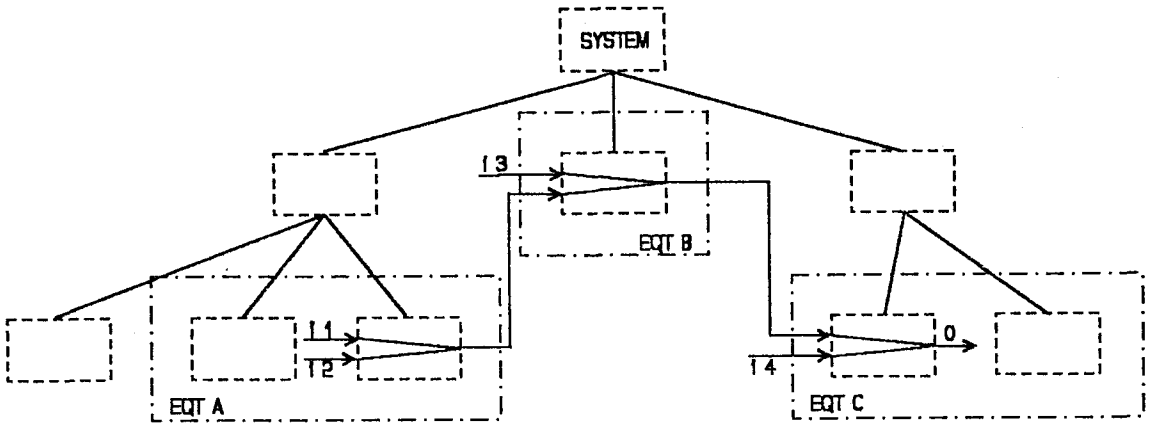
This phase consists of fitting the functional architecture previously defined within the hardware architecture suggested during the preliminary definition. The functional modules are shared among the equipment and identified as hardware or software.

This allocation is defined according to compromises taking into account factors like bus technology, real time optimization, dependability, industrial organization, etc...

Because of the allocation of the modules in the equipement, the functional data are broken down into three categories :

- inter-equipment digital data which will be processed and formatted to form the data bus messages and frames.
- inter-equipment analog data which will be processed in order to elaborate the analog wiring diagrams
- data exchanged between modules inside a single item of equipment.

The architecture is validated through the set-up of functional strings. As the functional graph provides a vision of "treatment oriented top-down breakdown", a vision of "recomposed data flow lines" corresponding to each operational function is needed to optimize data transformation through the numerous functional modules. It is therefore possible to extract "data functional strings", from the architecture data-base in order to study performances such as time delay or safety (construction of fault trees).



Data functional string for output O = f(I1,I2,I3,I4)

Formal tools

- General purpose tools
- Building the architecture graph is a task supported by OCS tool. This tool allows an aided graphic design for the architecture, the coherent collection of interface at any level of the break down and functional string identification.
- Integration of the functional architecture within the hardware architecture is supported by OEA tool. This tool allows, starting from the functional data collected with OCS, to determine automatically the load of buses connecting equipment, corresponding to a given allocation. By this way, it is possible to compare different hypotheses of allocation in order to optimize the repartition of functional modules in the equipment.
- The inter-equipment digital data, will be processed and formatted using GIN tool, in order to complete the data bus messages and frames.
- The inter-equipment analog data will be processed using SINOPTICS tool, in order to elaborate the synoptic wiring diagrams.

Simulation tools

- Nil

5.4 - Detailed definition

The detailed definition step consists of the set-up of the contractual documentation, covering hardware and software aspects needed for equipment manufacturing.

This documentation is made up of :

- Equipment integration technical specification (ITS)
- Detailed functional specification (DFS).
- Digital interface specification
- Analog interface specification

Integration technical specification documents

These documents describe technical hardware characteristics of each item of equipment (physical, electrical and mechanical ones) as well as its autonomous functions performed either by hardware of equipment software (system software excluded).

Detailed functional specification documents

The detailed functional specification step consists in producing the specification of the transfer functions of each functional module identified in the architecture.

Ideally, these documents contain "all that is necessary and only that which is necessary for the production of software and hardware" and therefore represent a definition with regards to the producers. The modularity of this documentation (an autonomous document per functional module) induces a sharp production constraint, since the software architecture of each item of equipment must comply with the breakdown imposed by the functional analysis of the system.

The language used in functional specification description may be natural or formal according to the nature and the quality requirement of the specification (e.g. critical or non essential function, company experience on the subject...).

DFS are the hinge between the system activities, under Avions Marcel Dassault responsability and software activities spread over numerous sub-contractors.

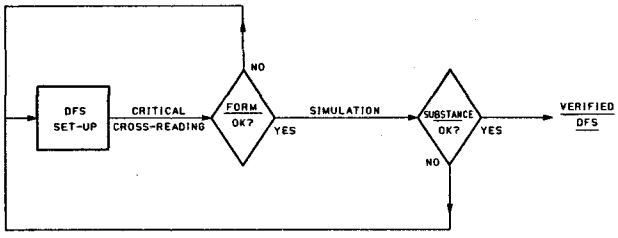
As they represent a very large contractual documentation, a big effort is made to improve their quality through a heavy verification process, using rapid prototyping.

DFS Rapid-prototyping

The DFS rapid-prototyping consists of three steps :

- critical analysis of the DFS
- simulation achievement
- DFS verification using the prototype

These steps occur in the following sequence :



- The goal of DFS critical analysis is to improve specification quality from a formal point of view, i.e. get the assurance that the specification documents are :

  . readable
  . coherent
  . complete
  . without ambiguity
  . feasible from a software point of view

  As a specificator is naturally satisfied with his document because of his knowledge of operational issues, it is necessary, in order to get valuable results, to keep the critical cross reader unaware of these operational concerns. The order given to him is : do not judge what the specification should do, but judge how it is written and how you could use it to develop software. As a consequence, the cross reading team is distinct from the specification team and is kept away from any operational requirement.

  Errors detected during the critical analysis steps belong to three main categories :

  . rigor errors
  . generality errors
  . specification inadequation to a software design

- The simulation achievement consits of two phases :

  . the simulation's architecture design
  . the coding of treatments specified

  To ensure a maximum representativity, the simulation's architecture is the same as the system functional architecture. Though, all the interface collected at the functional analysis step are transfromed into software data (Fortran data in this case) through an automatic process.

  This data base is used as a "common" for the DFS's simulation ; these DFS are simulated based on the transfer functions they contain. Such as to reach the following correspondance : one functional module = one DFS document = one simulation software module = one embedded software module.

- After a debugging phase, the prototype is then put at the disposal of specfiicators, for DFS verification. This verification is achieved using test scenarii applied to each module, then to the complete set of modules. This testing work is done as well by the specificators themselves, as by other engineers involved in the development. A specific interactive software allows the functional interface stimulation and the statement of changes during the simulation process sequences, combined tests (to look for the transition of a specific given output) or random tests (when combined can be applied).

- What the rapid prototyping returns :

  . specification quality improvement : the form control helps to correct and complete the document to raise it to a convenient level, allowing direct coding by programmers ; it acts as a powerful error revelator and an efficient quality control. Through the use of the prototype, it is therefore possible to verify accurately the detailed specification at the module, equipment, sub-system or system level, either in a "horizontal" way (exhausitive tests of one item) or in a "vertical" way (test of a given functional string).

  . The completion of a library of static and dynamic tests which will be re-used during the validation steps (rising branche of the "V").
  . A living functional reference, available before the equipment realization and which can be use for pedagogical or didactic applications, as well as to analyse and check the system operating or to define further modifications.

Formal tools

- General purpose tools
- System design tool : OCS
- Specification tools :
  . SCRIBE : textual specification in natural language
  . MITIA : specification of graphics
  . CISELE : formal specification for critical functions

Simulation tools

- Rapid prototyping tool : OCCAM

5.5 - Development of software and hardware

The specific activities of hardware and software development follow the system analysis described above.

The methodologies for software development as well as corresponding workshops can be slightly different according to producers.

However, the industrial architect of the system will make sure they fit in the general system development methodology, by audits or through the use of adequate standards.

5.6 - Equipment functional acceptance

The purpose of this step is to proceed to a separate evaluation of each item of equipment in order to make sure of a sufficient level of quality before integration of the system.

The step is divided into two phases :

1st phase : (in the manufacturer's factory)

The principal consists in achieving functional scenarii at the equipment level, in order to :

- validate the functions implemented in the equipment, relative to the specification (DFS + ITS)
- measure various characteristic operating margins (drawing time for visualisation equipment, memory and computting loads for computers, etc...)

The test scenarii used come from several sources :

- functional scenarii coming from DFS rapid prototyping
- scenarii generated by means of the system integration bench when it is available
- in-flight recorded scenarii through the use of test installations.

All these scenarii, after formatting, are applied to the under-test equipment through a standardized mean : MAGE, which allows the electric and digital interfacing of equipment.

2nd phase (in the aircraft designer's factory)

- in case of equipment with large system added value :

  For the equipment implementing the larger part of system software ("heart" of the system), such as mission computers or man/machine interface management equipment, representativity and test covering obtained by the use of MAGE is not sufficient.

  These item of equipment are tested within a specific environment representative of the whole system operation : the Hybridable Integration Center (HIC).

  From a hardware point of view, the HIC is constituted by :
  . a real time data processing center, (using non specialized and powerful computers), which supports a model of the avionic system.
  . a simulated cockpit linked to an external image restitution device
  . work-stations for test operating

  From a functional point of view the model is obtained by putting together several items of simulation :

  . a global simulation of the avionic system using the DFS prototype set-up with temporal and functional characteristics allowing direct interfacing with real equipment.
  . simulation of the external environment of the system, composed of representative models of aircraft sensors and operation theater.

  Besides the aid for equipment validation, the HIC allows the global verification of the design steps, users having the opportunity to compare, from an operational point of view the system's detailed model available on the HIC to the previous general model available on OASIS tool.

- In the case of sensor equipment :

  The sensor equipment validation requires the use of appropriate integration benches, allowing to check their performance by response to stimuli specific of sensor physics. Such benches are available for radars, inertial sensors or air data sensors.

  These benches can be connected either to the HIC or to the global system integration bench.

Formal tools

- Multi-purpose tools

Simulation tools
- MAGE
- HIC

5.7 - System integration/validation on benches

The purpose is to make sure, before integration into the aircraft, that the system is in compliance with its definition, and to evaluate the behaviour of the system through all its operation aircraft domain.

The job is done on the system integration bench. The environment of the airborne avionic system is simulated on the bench using either simulation or stimulation technics.

- Stimulation consists in replaying a scenario recorded on aircraft on the bench in order to set the system in a state identical to that encountered in flight.
- Simulation consists in producing an interactive scenario, thus allowing the piloting of the system and studying its answers in all its operation domain.

The integration benches nowadays deal with most of the validation and qualification tasks of the systems, reserving the flight tests for a few critical tests for which the bench is not sufficiently representative (particularly for physical environment, pilot workload, etc...).

Their advantages are their low cost and flexibility of use (compared to those of an aircraft), their availability, their ease of evolution, their present representativity and lastly their ever increasing analysis power. Furthermore, they are of easy access to the engineers.

5.8 - Aircraft integration/validation

The purpose of this step is to check and guarantee the operation of the airborne system. This activity includes ground tests and flight tests.

Integration tests are carried out for domain opening, store separation, complementary functional tests of the tests made on the integration bench and finally, particular operational evaluation on request of the aricraft manufacturer himself or his customers.

Various means are used, for example : prototype aircraft, test installations, data link devices, telemetry, real time software complex, mission preparation, read-out means and various ground support equipment.

6 - CONCLUSION

High Quality and performance is nowadays required for complex avionics.

The achievement of the objectives assigned to systems is done through a severe control of the development process with an emphasis on the requirement and specification activities.

The mastery is obtained by applying strict methodolgy, continuously covering the whole development cycle and which is to be the starting point for every Quality action

This methodolgoy is supported by formal and simulation tools grouped in a system workshop and software workshops.

The methodology summerized in this document has been progressively installed in Avions Marcel Dassault design offices since 1982.

Its full implementation through the use of the sytem workshop dates back to 1987 for the development of enhanced versions of MIRAGE 2000 (avionics up-date). At this time all formal tools and the simulation tools relative to the design steps were operating.

Today, the system workshop used in a multi-industrial context is installed for the development of RAFALE D aircraft and HERMES space shuttle, the systems of which being developped in cooperation with several aircraft companies.