Technische Universität Hamburg-Harburg

Department of Software, Technology and Systems

Project Work

Case Study: Validation and Verification of Cabin Core Systems for the Airbus A380-800 aircraft model

Submitted by:

José Alejandro Uriza Zepahua

Information and Media Technologies

30184

Supervised by:

Profr. Dr. rer. nat. Ralf Möller

Hamburg, March 3, 2005

Table of Contents

List of Figures

List of Abbreviations

Acronym	Description
AAP	Additional Attendant Panel
A/C	Aircraft
ACP	Area Call Panel
ADS2	Avionics Development System, 2nd Generation
AFDX	Avionics Full Duplex Switched Ethernet
AIP	Attendant Indication Panel
AMU	Audio Management Unit
ATA	Air Transport Association
BCEVI1	Electric Systems Integration & Test Cabin Management Systems
BITE	Build In Test Equipment
CAM	Cabin Assignment Module
CAN	Controller Area Network
CATEGA II	Computer Aided Test Generation Assistant (Version 2)
CIDS	Cabin Intercommunication Data System
CITR	Cabin Integration Test Rig
CMS	Central Maintenance System
CVT	Current Value Table
DEU	Decoder/Encoder Unit
DEESi	DEU Electrical Environment Simulation
DIR	Director

ECAM	Electronic Centralised A/C Monitoring
EPSU	Emergency Power Supply Unit
FAP	Flight Attendant Panel
FEDC	Fire Extinguishing Data Controller
FM	Failure Message
FSB	Fasten Seat Belts
FWS	Flight Warning System
GUI	Graphical User Interface
IBU	Integrated Ballast Unit (Cabin Light)
IDEFIX	Interface of Data Exchange in Test Facilities between IP and AFDX
IPCU	Ice Protection Control Unit
LDCC	Lower Deck Cargo Compartment
MMC	Maintenance Message Control
MPB	Multipurpose Bus
NS	No Smoking
OBRM	On Board Replaceable Module
OMS	On Board Maintenance System
OE	Original Equipment
PA	Passenger Address
PISA	Passenger Interface and Supply Adapter
PRAM	Pre-Recorded Announcement & Boarding Music
PTS	Purchaser Technical Specification
PTT	Push-To-Talk
S/D	Smoke Detector
SDF	Smoke Detection Function
SIB	System Integration Bench
TDS	Test Data Sheet
TIP	Test Input
VL	Virtual Link
V&V	Verification and Validation

1. Introduction

In critical information systems, the accuracy in the implementation of the system requirements is an important aspect to be considered. These requirements represent the functionality of the complete system that intends to be a software and hardware solution for a specific key problem. In order to assure the correctness of the requirements, diverse means of verification and validation in several levels have been developed in theory and applied in practice. This verification and validation means can be subject, as any other processes, to improvements in their performance.

The purpose of this project work is to present current industrial instances and solutions of verification and validation strategies for system integration in cabin core systems. Specifically the Airbus A380 Cabin Intercommunication Data System (CIDS) and the Smoke Detection Function (SDF) [1] are presented, providing suggestions of improvement for to the Airbus' process. An analysis of advantages and disadvantages of the testing strategy for every system is achieved by contrasting the theoretical proposals related to testing from literary sources with the actual state-of-art style of work in the critical information systems industry. After this analysis, suggestions of improvement will be formulated for each of the testing strategies presented. A final summary including conclusions and open problems is given at the end.

The motivation of this project work is to highlight the importance of system testing in the software development of embedded, real-time systems in a specialised branch, as the aircraft production is. It is of special interest to analyse a real instance of software testing in an innovative area of application, such as systems for aircrafts. The goals of testing are to find defects and to verify that the software meets its requirements. The basic levels of testing performed during the Software Development Lifecycle in order to achieve V&V activities are unit, module, integration, user or acceptance, and regression testing. In this project, the level of testing to be examined is the system integration testing.

Differences between the system integration testing level and other testing levels are remarkable, appearing in this project work. Compared to unit testing for example, the V&V activities in system integration level are performed by an independent external team [11], while in unit testing the same development team test their work themselves. The scope of the test cases is different as well, since in unit testing it can be designed to test specific code, individual modules or classes. On the other hand the test cases for system integration are more complex and structured, testing system functions at high level and from an external point of view.

As the software system being tested becomes more complex a more flexible test environment is needed. Test software packages called simulators, which perform in the same manner as some

piece of hardware or other software, are frequently used. In this project, the simulators verify the correctness of the implementation of the system requirements at a local level, before a complete integration of the cabin core systems with all the other systems in the aircraft. A description about the local simulation environment for CIDS and SDF will be given in this work.

As stated in [6] and [13] according to this testing level, the preferred testing technique for the system integration testing level is black box testing, also known as functional testing. Black box testing is “the application of test data derived from the specified functional requirements without regard to the final program structure” [14]. Given that the test team has no access to the code, but only to the software embedded in the hardware, black box testing is used in this project.

A general description of the testing process is found in the works of Lewis [11] and Schulmeyer [18]. As test design proceeds, the test plans are expanded into specific test cases, test scenarios and test procedures. After manual test execution, the documentation of the expected results is necessary so that actual results may be evaluated to demonstrate success or failure and act in consequence: report the problem, fix it and re-test. An organisational process is already defined and it is followed to support the testing activities in this project work.

This project work was developed with the support of one of the most important industries in Germany: Aeronautics. Airbus, a European integrated enterprise with facilities in Hamburg, Germany, is an aeronautic company focused in a leading aircraft manufacturing that consistently captures around half of all orders for airliners with more than 100 seats.

The A380 is the latest and largest very-long-range, four engine subsonic commercial transport of the Airbus family. This aircraft needs of critical information systems in order to have a full operability. Systems in the cockpit, cabin and cargo areas are essential to monitor and perform several important functions that keep the aircraft in operation.

Airbus has three main divisions: Operations, Manufacturing and Procurement. In the Operations area, the Cabin & Cargo Customisation department is in charge of the Cabin Innovation and Design Centre, Definition, Realisation, and Engineering Cabin & Cargo Customisation. This last one manages, amongst others, the department of Validation & Verification of Cabin Systems Integration. As sub department, appears the Cabin Core Systems department. It is in this unit, also known as BCEVI1, where this project work is realised.

2. Common Validation and Verification Considerations

Before the presentation of the case studies for each system, it is necessary to understand the common framework where they are tested. CIDS and SDF are actually complex real-time systems composed by software & hardware and both cabin systems interact in a very cohesive manner with other peripheral aircraft systems. Because of these reasons, it is important to introduce this framework previously to a further analysis.

The framework is the common basis for the validation and verification activities in system integration. The complexity inherent to the integration of the cabin systems sometimes prevents an automated testing, which is typical in unit test cases (e.g. Junit [7]). Besides complexity, other reasons that support the decision of executing manual testing during this project work are the complicated configuration of the test environment, the time restrictions for automation development and the limitation of the available testing tools. However, as it will be seen, the tools used during this project are a very useful aid in order to monitor and validate the system behaviour in a real-time scenario.

Regarding the complicated test environment configuration it is important to mention that an elaborated common test platform has been built up; defined as System Integration Bench (SIB). SIB is necessary because eventually the mutual interaction of both systems should be also verified. The interaction is caused because both systems reside in the same central control unit, the so-called Director (DIR). The following sections will describe briefly how the SIB platform is built and how the different simulation and monitoring tools are integrated to the SIB. In the specific case study for each cabin system, the initialisation of every test is explained in detail.

The mentioned framework is made up both by theoretical and technical components. The theoretical components are the System Integration Testing Techniques and the BCEVI Test Process. The technical components are the Test Platform Configuration and the Main Simulation Tool. Both components are complementary, since both aspects are important to establish the process background (theoretical component) and support the complex execution of the testing (technical component). An overview of both kinds of components will be presented in the following sections.

2.1 Theoretical Components

As explained before, these components are the basis that supports all the V&V activities performed in the Airbus testing department. An important component in this section is the general description of author's testing methodologies and techniques about black box (functional) testing, which will be compared at the end of this paper with the current testing activities performed in Airbus. Such comparison will provide suggestions for improvements in the current testing activities.

The Airbus' proprietary testing process is also described in order to learn the necessary activities to be executed in order to generate the testing products. This process is institutionalised across Airbus and establishes the methodology and V&V process for software system integration testing. A detailed description of this process will be provided in this chapter.

2.1.1 System Integration Testing Techniques

Several authors have proposed a series of techniques whose approach is to help in the software verification and validation process [14]. In this section the main black box techniques will be discussed and how their different perspectives provide support to the V&V activities.

First at all, as a general technique, the simulation has proved that is highly useful in the V&V tasks. Software verification is performed by determining, with the use of simulation, whether the model of the software behaves as expected on models of the computational and external environments. Simulation is most often employed in real-time systems development where the "real-world" interface is critical and integration with the system hardware is central to the total design. Balci [2] establishes also the importance of the validation of such simulations, in order to obtain a reliable platform to test the actual software.

For specific black box testing techniques, the most obvious is exhaustive testing. This is infeasible, since the domain of a program is generally infinite and cannot be used as data set. However, test data should be derived from the functional requirements and include both valid and invalid inputs. In this context, the Boundary Value Analysis (or Stress Testing) is a technique that assists to determine the input values that are in the frontier limit between data validity and invalidity.

A second technique, Design-Based Functional testing, is based in the construction of a function tree where a requirements function is the node of such tree and the design functions are the branches of this node. The purpose of this type of black box testing is to derive the test data also from the design phase, not only from requirements phase. In this way, deep and more detailed test coverage is obtained.

Cause-Effect Graphing is a technique for developing test cases for programs from the high-level specifications. In this case, a program with *n* possible stimuli (called *causes*) will generate 2^n inputs. Instead of generate this amount of test cases, a more sophisticated approach is to use the program specifications to analyse the program's effect on the various types of inputs. In this way, a decision directed graph is used to determine what causes and effects are related each other and reduce the test data to these input values.

Another categorisation of black box testing is related to the coverage of the test cases, as presented by Quirk [13]. Positive Testing is related to the test cases that are in charge of testing the valid inputs for the system, in order to verify if they comply with the system requirements. On the other hand, Negative Testing is related to the test cases focused in the input of invalid data, or those that will guide into a malfunction of the system. In this case, the expected result is an error and recovery management by the program. In both options, the analysis of the output will provide enough information about the validation of the software.

2.1.2 BCEVI Test Process

In this section, an overview of the Test Process followed by the A380 cabin system testing team will be provided. Any generated testing work product is known as Test Object. These Test Objects are generated in every phase of the development. In the Design phase of any System to be tested, one activity to be done is the definition of the Test Execution Plan (TEP). This definition includes the Test Items (TI), those are the minimum testing units depicted by the Test Features (TF), which are the testable characteristics of the system. The System Requirements (SRQ) are bounded to the TF since one or more Test Features can cover these requirements.

Test Objectives (TO) are defined in the Test Design (TD) phase. TOs describe the “path” for the upcoming test. It may define the possibility of executing the Test in different forms and types within a TO. Test Objectives include several of the Test Features identified in the Design phase.

As part of the TO, the Test Requirements (TRQ) are included and linked to the System Requirements defined in the Design phase. A TRQ defines a Test or Test section to be fulfilled. Possible TRQs are preparation, interface check, calibration, etc. They can (but not necessarily) be included directly in the SRQ.

The TRQ includes a depth that describes the granularity of details of the Test. Here can be differentiated the details of the test. It is focused mainly in automated tests, thus it is not applicable for the tests performed for these real-time cabin systems, CIDS and SDF. The TRQ also includes the Tag, which is the smallest executable unit of one test. It would describe the instructions directly, e.g. definition of execution date and time.

The last phase, Testing, is relative to the formal execution of the designed tests. A level (LV) of importance is assigned to the test, from LV0 (validates if the system works) to LV4 (validates the variations of the systems). Test Cases (TC) are prepared with the information defined in the TRQ, TF and TAG, describing how the test should be performed. Test Procedures (TP), including several TCs, are structured and represent the guide for the actual Test Execution (TE). All the previously described Test Objects are created by the Test Designer, one member of the Test Team.

The Test Executor, another member of the Test Team, retrieves the information about the TC and TP and performs the testing, creating a TE object. At the end of each test execution, the results are compared with the descriptions contained in the TC and the tag. If according to this comparison, the resulting behaviour of the TE does not match with the expected behaviour described in the TC and TAG, then a Problem Report (PR) is raised.

A summary of the relationships between the Test Objects is provided in this diagram:

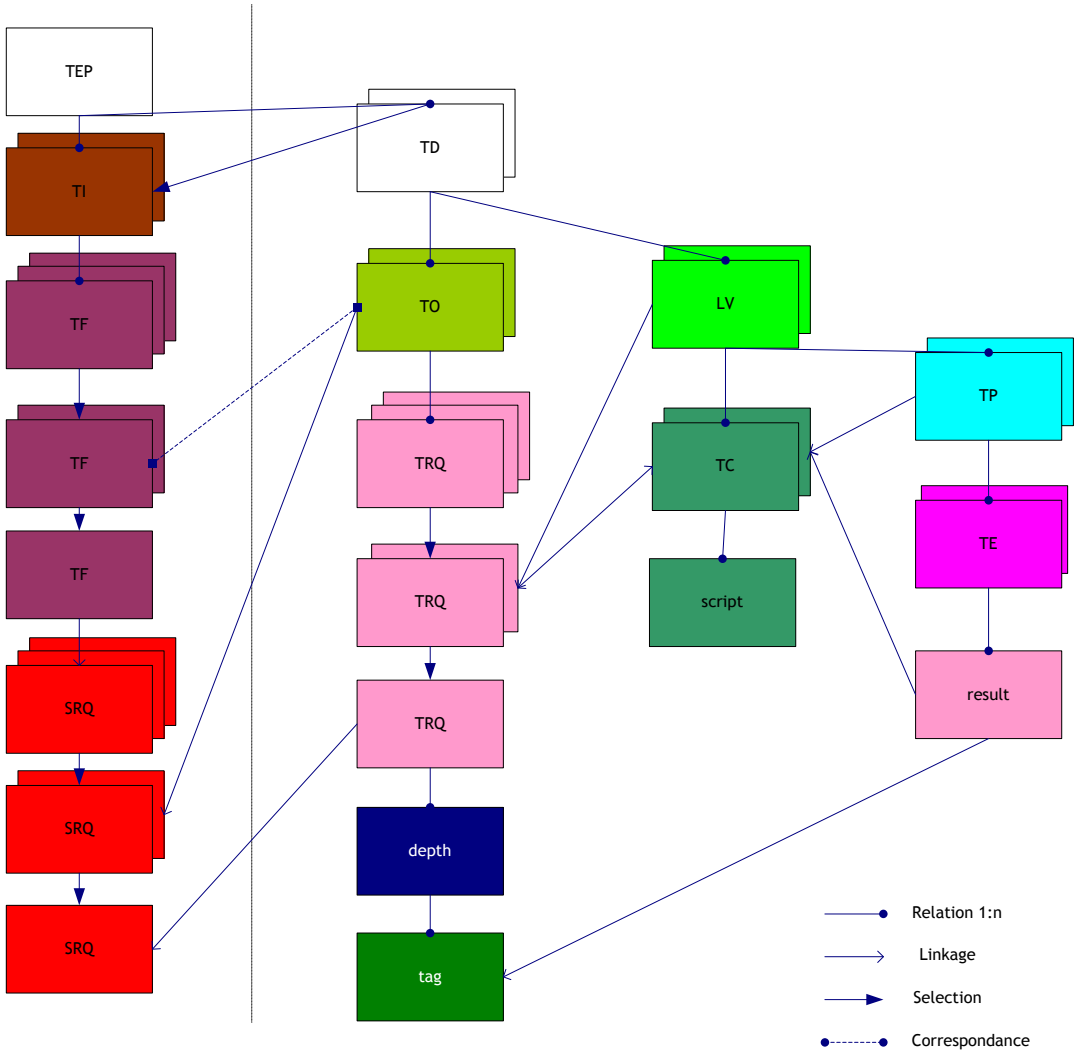


Figure 2-1. Object Relationship in Test Plan

The following chart describes the flow of communication between the different teams (Design, Development and Testing) during the testing phases described above. The Design team submits Issues, which are the documentation with the description of the system design specifications to the Development team, and TEP for the Testing team as described before (during the Test Design

phase). The Development team releases the software in a Cabin Assignment Module (CAM), an On Board Replaceable Module (OBRM) or an update of the software in the Director. Finally, the Testing team configures the new software releases within the SIB environment, producing a Test Result and, if necessary, PR that will be submitted to the Design team. Then they decide where the problem resides and consequently fixed. Depending on the solution, a new Issue (the problem was in the System Design) or a new software version (the problem was in System Development) is submitted and then the cycle continues.

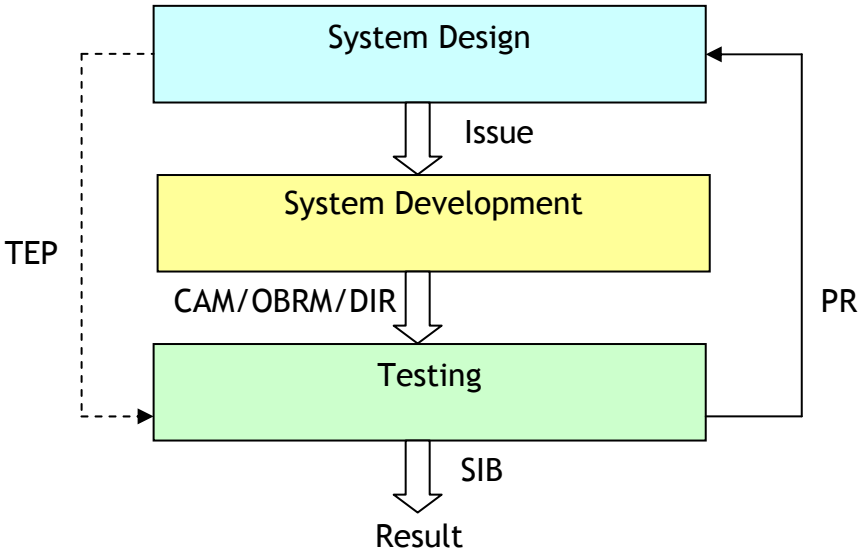


Figure 2-2. Test Process Communication Flowchart

2.2 Technical Components

The Technical Components are the tangible assistance for the testing activities. Both components were selected and allocated by Airbus after making several decisions. The main aim for selecting these components was to support the manual execution of the tests with possible automation support, as well as the simulation of the external applications surrounding the cabin core systems.

The decisions to set up the Test Platform and the Simulation Tool in this manner were based on the previous experience with the system development of other aircrafts. Test Platforms already exist in Airbus Deutschland to testing and provide maintenance to the systems of previous aircraft models. However, since this is an improvement of the previous cabin core systems for a new aircraft, the simulation tool is a new development from the supplier and contains features specifically designed for the enhancements of CIDS and SDF for the A380.

The Test Platform used during this project (including the software, hardware and the simulation tools) are intended to verify and validate the correctness of the cabin core system requirements in a local scope. That means that CIDS and SDF will be validated in an isolated manner, previous to a complete integrated test with the real peripheral systems. This global system integration test is performed in Airbus in the Cabin Integration Test Rig (CITR), which is out of the scope of this project work.

2.2.1 Test Platform Configuration

The System Integration Bench (SIB) is a platform with specific configuration and testing environment according to the software design specifications for the cabin core systems. This includes a set of simulation hardware and software infrastructure, as well as power supply installations in order to perform integration and system tests for CIDS and SDF systems. The complete description of the CIDS SIB is detailed in an internal Airbus document [10].

The A380 aircraft has an innovative configuration, including two passenger decks (Main Deck and Upper Deck) and one cargo deck (Lower Deck). The SIB disposes the distribution of the hardware simulators and power supply of the Cabin Systems in the mentioned aircraft locations, but in a reduced composition in order to facilitate the execution and monitoring of the testing. The current configuration of the SIB is presented in the following arrangement:

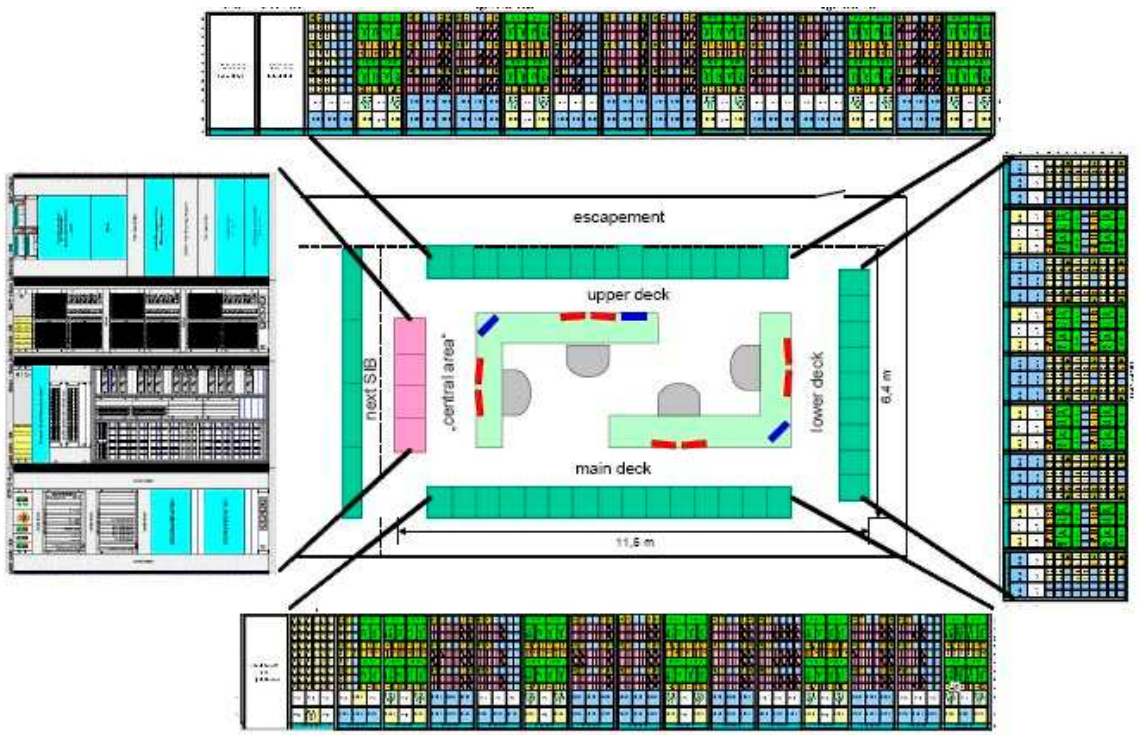


Figure 2-3. CIDS/SDF SIB Layout (from [9])

For the interaction of both software and hardware simulators with the system under test (CIDS or SDF) known communication protocols are used within the Test Platform. AFDX, Ethernet and CAN are the protocols that will support the message transmission and reception between the Directors and the simulators. The stimulation of the cabin systems is realised in the SIB using simulators of the electronic bay related systems (using Avionics Development System, ADS) and simulators of the cabin related systems (using Decoder/Encoder Unit Electrical Environment Simulation, DEESi). In addition, a real-time infrastructure is set to perform as the operating system of the complete Test Platform.

The central area of the SIB includes four cabinets: The first cabinet with the main power panel and the ADS/AFDX router. The second cabinet consist of the Gateway Director Audio, the Attendant Indication Panel and Area Call Panel (AIP/ACP) simulators in the cockpit, the Ethernet router from ADS network to the Director in the SIB, and the three Directors' circuit breakers for both power supply (Normal and Essential). The third cabinet includes the Connectors Panels for every discrete signal of each Director (1, 2 and 3). Finally, the fourth cabinet hosts three multimeters for verifying the output of the signals in the panels in third cabinet, and also holds three trays, one for each Director.

The Upper Deck area of the SIB includes 16 cabinets: In the Server cabinet are located several routers for the AFDX, Ethernet and other networks, and one Real-Time Tester Cluster. In the second Server Cabinet are located the Flight Attendant Panels (FAP) circuit breakers, the ADS Virtual Machine, the Banz server for DEESi, the Configuration Server and the Data Server. There are also 4 cabinets for Doors that allocate simulators for AIP/ACP, Discretes B, Additional Attendant Panel (AAP), Smoke, Temperature, Handsets and Decoder/Encoder Units type B (DEU-B). Additionally this section includes 9 cabinets for Seat Rows (one of them also for Lavatories and Stairs), allocating simulators for Illumination Ballast Unit (IBU), Discretes A, Stand Alone Passenger Interface and Supply Adapters (StA PISA) and Decoder/Encoder Units type A (DEU-A). Finally there is one cabinet for Galley, Aft Lavatories and Stairs, with DEU-B interfaces.

The Main Deck section of the SIB includes 16 cabinets: The A/C power supply (normal an essential) cabinet with DC and AC displays, and the Upper Deck, Main Deck, Lower Deck and Central Area switches. The DEESi power supply cabinet with switches and indicators for Upper Deck, Main Deck and Lower Deck. It also includes 5 cabinets for Doors as described in the Upper Deck area, but one of them also is configured with Aft Lavatories and Stairs and therefore additionally including simulators for IBU, Discretes A and DEU-A. There are also 9 Seat row cabinets as described above, but one of them for Forward Lavatories and Stairs. Finally, the Cabinet Gateway for Cargo Smoke includes the Smoke Hardware simulators, the Squib simulator, the Fire Extinguishing Data Controller (FEDC) Hardware simulator, and the DEESi simulators of the Smoke CAN buses and FEDC.

The Lower Deck section of the SIB is still under construction at the time of this project work, but it is supposed to include also DEU-A, DEU-B and mixed DEU-A/-B interfaces with specific simulators according to the original equipment requested in the A380 final aircraft configuration for this deck.

Before any testing occurs in the SIB, DEESi should be configured. To this purpose, a start-up script is run containing the commands to assign values to the simulators in order to setup the environment. This is necessary since the default values loaded into the hardware simulators have to be adequate to the specific test description.

There are several tools that are configured and ran in the terminals in the mid area of the laboratory. These tools can be categorised as Documentation, Simulation and Monitoring tools, based on its utility to the execution of the testing. The Documentation tool is mainly used by the Test Designers in order to create the Test Objects, and also by the Test Executors to retrieve the Test Objects, document the results and raise Problem Reports given the case. The Simulation tools are configured and maintained by the SIB Support team, who are the people in charge of the organisation and functionality of the Test Platform. The Monitoring tools are used mainly by the Test Executors in order to verify and validate the correct behaviour of the system when a test is being carried out.

The Documentation tool is CATEGA II that is an Airbus proprietary tool used for the creation and access to the Test Objects, the documentation of the hardware configuration of each test in the SIB and the recording of the problem reports.

The Monitoring tools, used by the Test Executors are:

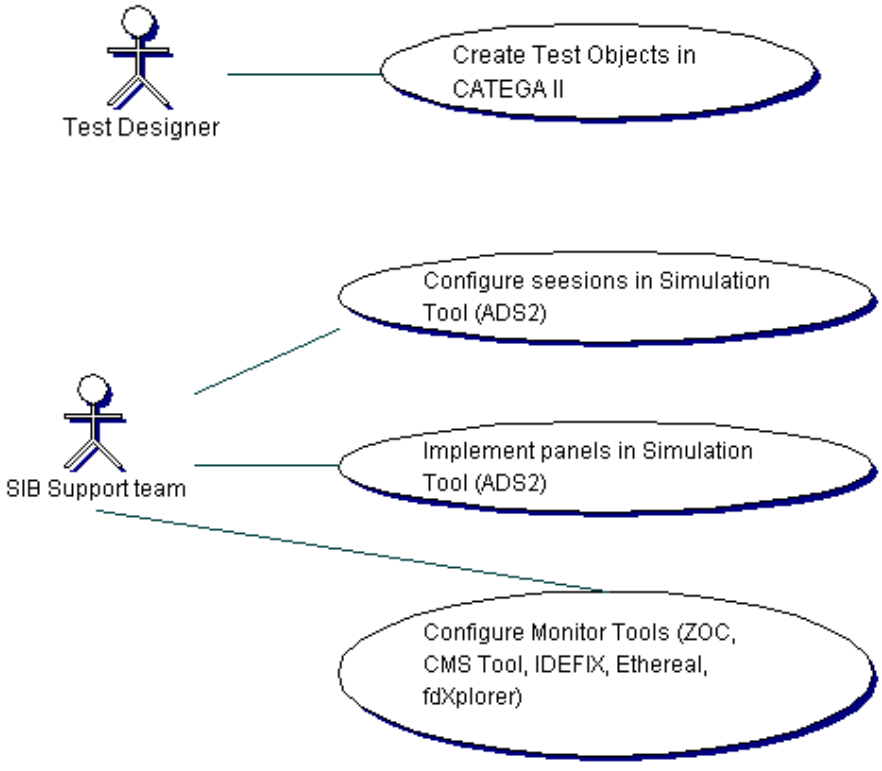
- ZOC: Monitor connected by the serial port to the director, whose purpose is to receive the messages of the Maintenance Message Control (MMC), reviewed during the test execution.
- CMS Tool: Home developed tool by Airbus France that allows the monitor of Built In Test Equipment (BITE) Fault Messages present in the network, sent by the directors. It also permits the modification of some variables of the flight, e.g. the flight number, flight phase and the pair of arrival-departure cities. This tool is not only a testing support tool, but also a production tool incorporated into the actual cabin systems in the aircraft.
- CANalyzer: Development tool for CAN bus systems with whose help data traffic can be observed on the bus line, analyzed and supplemented.
- Ethereal: Monitoring tool that overviews the traffic of messages in the Ethernet network. It is helpful to verify that the messages are correctly send and received between the DIRs and the other components of CIDS and SDF.
- fdXplorer: Terminal that monitors the traffic of the AFDX messages. This terminal is very useful when supervising the raw messages during a specific period of time, and then provides information to be used during test results analysis, both for CIDS and SDF.

The Simulation tools, configured by the SIB Support team and used by the Test Executors during the execution of the tests are:

- ADS2: Tool for simulation of the aircraft avionics environment. It will be explained in detail in section 2.2.2
- ASCIIBanz: Command-line user interface for configuring the simulators of DEESi for each cabinet in the Upper Deck, Main Deck and Lower Deck of the SIB as described above.
- IDEFIX: Web-based tool that allows the interaction between the BITE Messages from CIDS and SDF towards the CMS Tool, translating Ethernet frames to AFDX messages and vice versa.
- PuTTY: Terminal emulator that allows the connection and work in the Linux terminals from a PC environment.

Additionally, other components are necessary in order to execute the tests. There are two FAPs and one Mini-FAP that allow interaction in a Graphical User Interface level with CIDS. This equipment is manipulated through a touch screen and several buttons related to the operation of the system.

The following use cases explain the interaction of the test team with the tools mentioned above. For a detailed explanation about the description of the use cases, please refer to the sections “Current Testing Scenario” in the Case Study chapters for CIDS and SDF of this document.



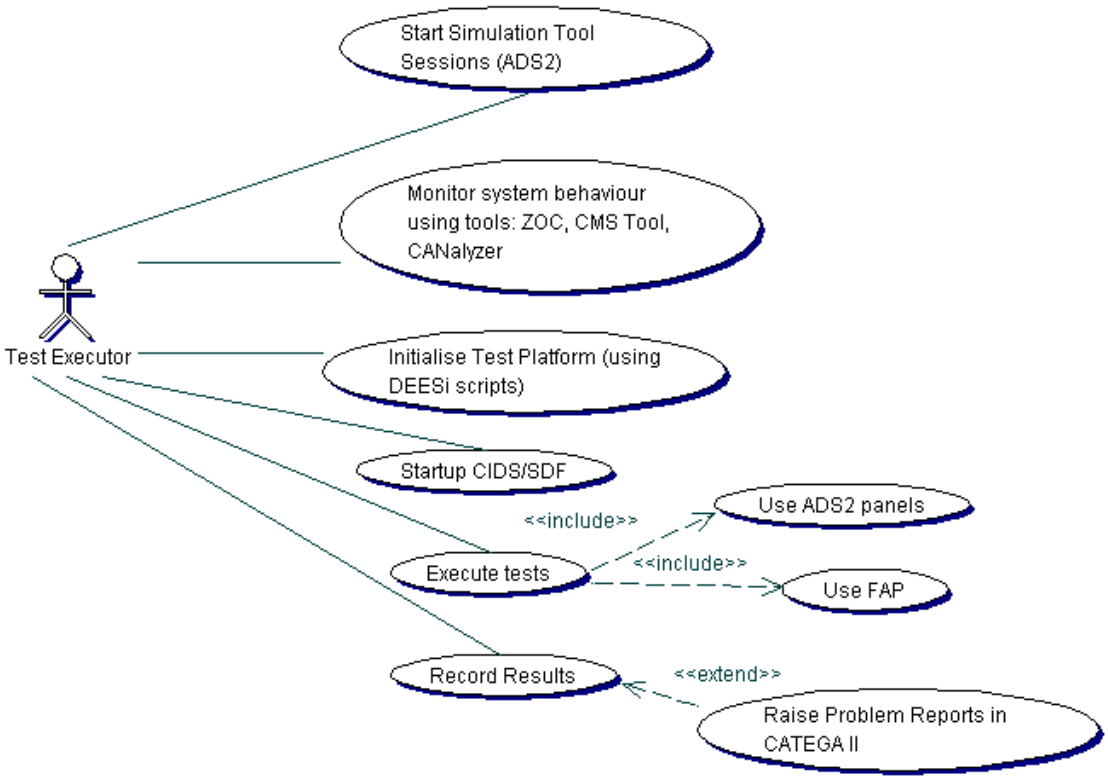


Figure 2-4. Use Cases related to Tools

2.2.2 Simulation Tool

In this section only ADS2, the Avionics Simulation Tool, will be presented. ADS2 stands for Avionics Development System 2nd Generation. It is a real-time test and simulation environment specifically designed for test, integration and validation tasks in aerospace applications. The detailed description of this tool was provided by Tech S.A.T. [19], the tool supplier.

This simulation is essential because the cabin systems communicate with all the electronic components and systems of the aircraft providing information from the cockpit and other external structures. Given that the validation of CIDS and SDF are only in a local scope, all this interaction should be simulated using ADS2, a tool that provides these stimuli.

An ADS2 system is comprised of the following components:

- Physical hardware, including crates, I/O Cards, connectors, workstations, etc.
- devRTCore – the distributed real-time core software system
- Installation-specific, low level support software for actual hardware
- High level real-time tools set.

The main interaction point of this tool is the Session Configurator window, as seen in the figure below. In this main window, sessions can be defined that represent the minimum set of features needed to run an ADS2 simulation. The sessions include all the necessary components and panels to be used during the testing simulation. In order to run a simulation, the corresponding preconfigured session is loaded in the ADS2 tool, and then the environment is ready to start the testing.

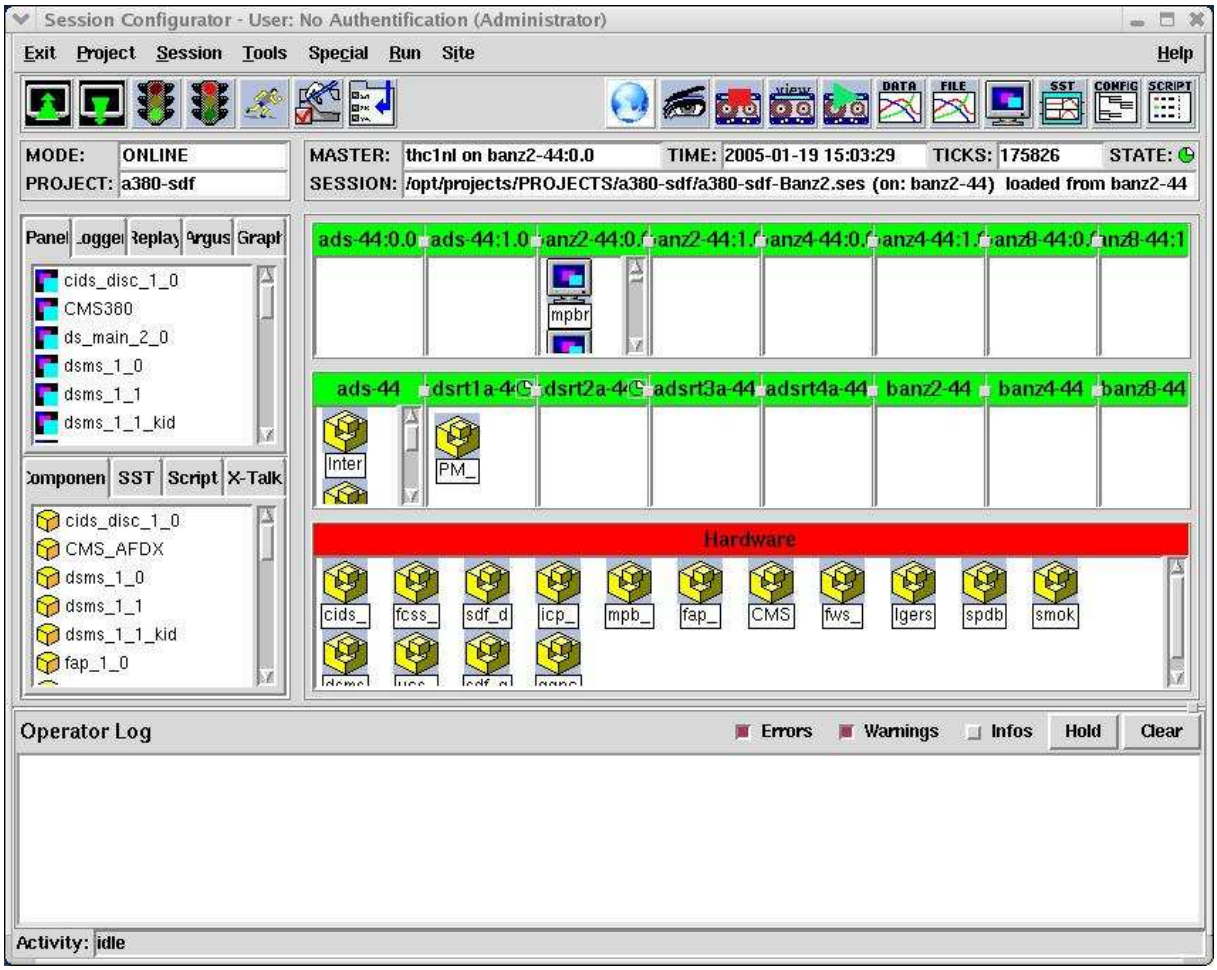


Figure 2-5. ADS2 Session Configurator

The basic concept of the ADS2 software is the Current Value Table (CVT). The CVT is a distributed, real-time database holding all dynamic data of an ADS2 application. devRTCore provides access methods to CVT variables that can be used by any task on any networked computer in an ADS2 system.

The I/O subsystem consists of the generic I/O process, device drivers, and the various I/O hardware resources (boards). It is responsible for processing data from input devices and storing it in the corresponding CVT variables and for sending data from selected CVT variables to the corresponding output devices. There are no direct interfaces between tools and I/O devices. All data

exchanged between the various ADS2 components flows through CVT variables. The data source of a CVT variable can be: an input device, a panel, a simulation, a stimulation setup or a test script. Data consumers for a CVT variable can be: output devices, panels, simulations, etc.

In this specific configuration for the A380 Cabin Core System Testing, the SUT (System Under Test) is the software that is loaded in the three Directors that are part of the CIDS/SDF architecture. The three Directors are the central components of the CIDS. One director is in active mode, and the other two are in hot standby mode. All the components of the CIDS are connected to three identical directors. The connections are realised by data-bus interfaces and discrete interfaces. Furthermore CIDS has interfaces to other systems.

The CIDS/SDF components are connected to each other and to the other systems by the following types of interfaces:

a) Discretes: The discrete interface is a unidirectional interfaces which transmits a 28 VDC or ground signal.
b) Audio: The audio interface is a unidirectional interface. It transmits signals between 50 Hz – 11 kHz if connected to a DEU type A and 200 Hz – 5.5 kHz if connected to a DEU type B.
c) Data bus: The data bus interface is an unidirectional or bidirectional interface:
    a. Unidirectional interface: CIDS uses RS232 unidirectional interface for data transmission for EPSUs, AAPs and AIPs. This data bus transmits 8 bit data words with 9600 B/sec.
    b. Bidirectional interface: CIDS uses the CIDS buses (Top Line and Middle Line), Ethernet, AFDX and CAN for bidirectional transmission.
        i. Top Line and Middle Line: These buses transmit data between directors and all DEUs. The buses transmit 14 bit data words with 4 MB/sec.
        ii. Ethernet: This bus transmits data words with 10 MB/sec.
        iii. AFDX: This bus transmits generally data words with 10 MB/sec, same as Ethernet.
        iv. CAN: This data bus transmits at a baud rate of 83.33 KB/sec.

In order to simulate the interfaces mentioned above for testing purposes, several panels are implemented in ADS2 allowing the interaction between the user and the real-time avionics simulation. These panels allow the injection of data into the simulation tool using buttons to switch on signals, and also permit the visualisation of the signals in a red-green schema, to indicate when a signal is present (green) or not (red) in the simulation environment. The purpose of the panels is to permit the Test Executor to apply stimuli to the system under test within the simulation environment, according to the specifications written in the Test Cases.

These panels are implemented using Tcl/Tk language, allowing the interaction with the CVT through a GUI. In the corresponding sections of this project work, the panels used by for each system will be presented and explained. An example of a panel of discrete signals coming from the cockpit is presented in the following figure:

Figure 2-6. Panel of Discretes –realised in Tcl/Tk

ADS2 is a reliable tool that supports in a large extent the development of the test execution. As explained in [2] and [16], it is important that the simulator models would be also validated, verified and certified. This internal V&V is executed by an internal team from BCEVI1 too, the SIB Support team, who designs and implements the ADS2 panels as well as the testing of this implementation against the CIDS specifications and requirements documents.

One technique applied to validate the ADS2 panels is Event Validity, where the events of occurrences of the simulation model are compared to those of the real system to determine if they are similar (e.g. red or green signs in ADS2 panels vs. voltage measurements in Connectors Panels). Another technique applied is Operational Graphics, where the values of various performance measures are shown graphically as the model runs through time, using the additional tools provided by ADS2.

The certification of the ADS2 panels also goes through some other validations such as Operational Validity, in order to determine if the simulation's output behaviour has the accuracy required. Using Dynamic Testing technique, messages tracing are evaluated with fdXplorer to verify the correct transmission of information over the AFDX network. Investigations about input-output relations through the voltage measurement on the Connectors Panels and the comparison of the results against the display in the ADS2 panels are necessary to support the accuracy of the implementation. Finally, the reprogramming of critical components guarantees that the implementation of the interface is covered in a large extent. This produces several versions of the Tcl/Tk panel, used to determine if the same results are obtained in all the several reprogrammed components.

Now that the basis of the testing infrastructure has been explained, in the following chapters the specific case studies about CIDS and SDF will be presented. The structure of the Case Study is conformed by the System Description, the Current Testing Scenario, the Advantages and Disadvantages of the Testing Strategy and the Suggestions of Improvement. These chapters are the main content of this paper, because they provide the facts and learned lessons about my experience during this project work in Airbus.

3. Case Study: Cabin Intercommunication Data System

This Case Study is focused in the presentation of one of the most complex systems in the A380 aircraft: the Cabin Intercommunication Data System. CIDS is an important system for the control of the functionality of the passenger decks in the aircraft cabin, whose compliance with the system requirements should be correctly verified. In order to fulfil this validation and verification request, a testing methodology based on the BCEVI Test process is followed with certain variants.

After the presentation of the system, the testing methodology is shown in two scenarios. These two industrial testing instances will be analysed in an advantage-disadvantage fashion and, in contribution with the testing techniques found in the literature, improvement suggestions will be formulated. With these elements, it is aimed to exhibit how the current business testing is working nowadays and how it is possible to enhance its testing techniques.

3.1 System Description

This system implements the ATA (Air Transport Association) chapter 44, as described in [1]. This chapter is related to Cabin Systems. Different systems are installed for communication between the cabin crew members, passengers and ground crew (e.g. maintenance). Air to ground communication is also possible.

The cabin core system has these functions: Cabin Intercommunication Data System (CIDS), Smoke Detection Function, Courier and Cargo Intercommunication, and Service Interphone. All functions of the cabin core system are done in the CIDS. CIDS is a microprocessor-based system. It operates controls and monitors the main cabin systems and can do different system and unit tests.

The different functions and the connected systems are:

- Passenger related functions:
    - Passenger Address (PA)
    - Passenger Call (Service Call)
    - Passenger lighted signs (No Smoking/Fasten Seat Belts (NS/FSB))
- Crew related functions:
    - Cabin crew interphone
    - Service interphone
    - Emergency evacuation signalling

- Cabin systems function:
  - General cabin illumination control
  - Boarding music
  - Pre-recorded announcement
  - Lavatory smoke warning
  - Temperature regulated drain mast system
- Programming-, monitoring-, test functions:
  - System programming and test
  - Reading- and work light test
  - Escape slide bottle pressure monitoring
  - Extended emergency lighting test

Additionally, CIDS has a Smoke Detection Function (SDF) with a separate power supply. This function continuously surveys the cargo- and the lavatory smoke detectors and monitors the cargo compartment fire extinguishing bottles. Further information can be found in the Case Study for SDF in this paper.

If the cabin layout is changed, it is not necessary to make a complex and time-expensive hardware change of CIDS components. To make it easy to change the cabin layout, CIDS hardware has spare inputs, outputs and circuits. These allow the connection of new and additional equipment without a hardware change of CIDS components. Furthermore the software of the CIDS defines all operations. If any equipment is changed, only the software database has to be modified, through the Cabin Assignment Module (CAM) to adapt e.g. the new cabin zoning. A system reconfiguration for the installation of options, cabin reconfiguration or CIDS expansion is thus decreased to software database changes and decreases the aircraft out-of-service time. CIDS is also designed to detect faults in CIDS components and in the connected equipment by itself. Thus scheduled maintenance is unnecessary.

The software defines all operations of the CIDS. This makes it possible to do system reconfigurations by software database changes, which reduces the aircraft out-of-service time. Many of the CIDS components contain comprehensive built-in test equipment (BITE) circuitry, to enable the CIDS to detect faults both in connected systems and in individual CIDS units.

The system philosophy is based on:

- A microprocessor-controlled data-bus system
- The connection to various cabin and avionics systems through different standard interfaces
- Sixteen to twenty-two data bus lines (twelve to sixteen top lines for passenger related systems and the cabin illumination and four to six middle lines for crew related systems)
- Three functional units for the data-bus control: the CIDS directors 1, 2 and 3

- One director in active mode and the second and third director both in hot-standby mode
- Immediate switchover to the second or third director, if a failure of the first director occurs
- An internal CIDS power management function to guarantee the relevant CIDS functions in relation on the A/C power status
- Independent touch-screen Flight Attendant Panels (FAPs) on the different decks of the aircraft to program, to control and to indicate the status of the CIDS and the related cabin systems
- Mini-Flight Attendant Panels (Mini-FAPs) installed near the attendant stations on the different decks of the aircraft to control certain functions of the CIDS and of related cabin systems
- Addressable Decoder/Encoder Units type A (DEU-A) for the interface between top line data buses and cabin related systems
- Passenger Interface and Supply Adapters (PISA) for the interface between Decoder/Encoder Units type A (DEU- A) and cabin related systems/units
- Stand Alone Passenger Interface and Supply Adapters (StA PISA) for the interface between Decoder/Encoder Units type A (DEU-A) and equipment/indications installed near the cabin attendant stations and the lavatories
- Light Interface Standardisation Adapters (LISA), which provide the digital interface between the Decoder/Encoder Units type A (DEU-A) and the cabin lighting devices
- Lavatory Illumination Adapters (LAILA), which provide the digital interface between the Decoder/Encoder Units type A (DEU-A) and the lighting devices in the lavatories
- Addressable Decoder/Encoder Units type B (DEU-B) for the interface between middle line data buses and crew related system components and door area related components
- Attendant Indication Panels (AIP) near the attendant stations on the different decks of the aircraft to display CIDS related messages
- Area Call Panels (ACP) installed in the entrance areas to inform the cabin crew about certain CIDS events
- Handsets at every attendant station to provide the cabin crew with the possibility to communicate with other attendant stations and the cockpit and to perform PA functions
- The definition of system properties and cabin-layout information in a software database stored on a standard mass memory card, the Cabin Assignment Module (CAM)
- Easy exchange of the CAM which is plugged into the FAP
- One On Board Replaceable Module (OBRM) which is plugged into the FAP and which stores the operating software
- One integrated Pre-recorded Announcement and Boarding Music (IPRAM) audio database plugged into the FAP. This memory card contains Boarding Music audio and announcement audio-files
- A Vacuum System Control Function (VSCF) to control and indicate the status of the vacuum toilet system and the potable water system

- Build In Test Equipment (BITE) to continuously monitor the performance of the CIDS and of the connected equipment
- A Smoke Detection Function (SDF) to indicate a detected smoke in the lavatories, in the cargo compartment and on the lower deck.

All components of the CIDS are connected to three identical directors through data-bus interfaces, discrete interfaces and audio interfaces. The same types of interfaces are used for the connection of external systems to the CIDS.

The following diagram illustrates the complete Cabin Intercommunication Data System architecture:

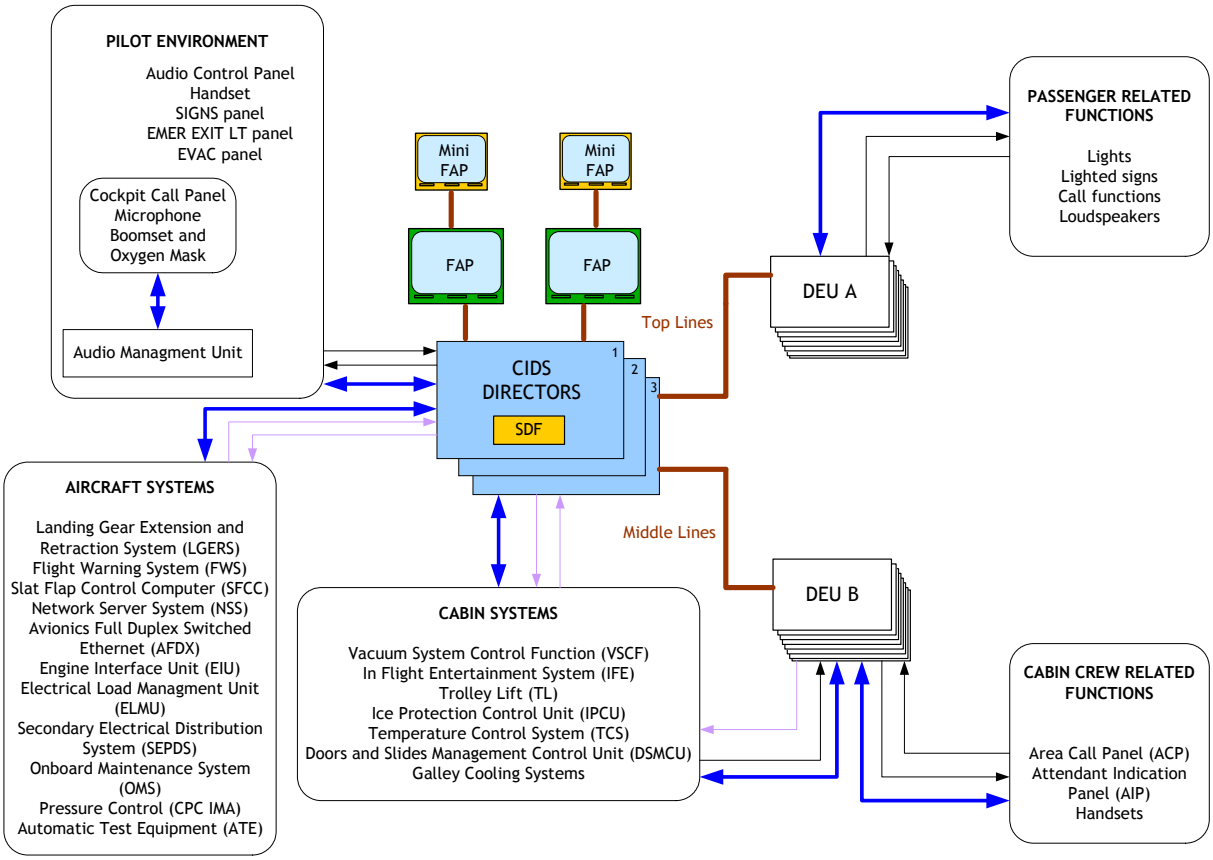


Figure 3-1. Cabin Intercommunication Data System (CIDS) Architecture

3.2 Current Testing Scenario

The testing activity of CIDS, due to its own nature of very complex system composed of several functions, was carried on by a team of four persons: one of them working in the preparation of the Test Cases as described in section 2.1.2 as Test Designer, documenting these activities in CATEGA II, and the other three persons executing tests in the laboratory, and raising and following-up

the PRs in CATEGA II. They are the Test Executors. This section describes two examples of how these tests were designed and executed in the scope of the BCEVI test process, and also illustrates the use of the tools described in the previous chapters.

The scope of the CIDS testing includes the following functions and features:

Function	**Feature**
Passenger Address	Direct PA from Cockpit / Cabin System
	PA under emergency conditions
	PA zone selection from Handset
	PA attention chime and monitoring
Cabin Interphone	Emergency call from cockpit
	Purser call from cockpit
	Normal call from cockpit
	Emergency call from attendant station
	Capt call from attendant station
	Call –reset, -divert, -intercept, party add, monitor
Service Interphone	Service interphone link with Emergency Call
	Service interphone links and Indications
	Activation/Deactivation/Override/Resetting
Emergency/Evacuation Signalling	EVAC CMD from Co-pilot and Cabin
	Request, Interlock, Takeover
	EVAC tone, visual indication and reset
	EVAC status transmission to other systems
Passenger Lighted Signs	No Smoking
	Fasten Seat Belts/Return to Seat/Cabin
	Decompression
	Dimming
Cabin Illumination	Decompression/Smoke detected event
	Main On/Off
Reading Lights	Reading Lights Function Local/Attendant Work Lights
	Reading Lights Function from FAP
Doors & Slides related indication	Doors
	Slides
Air Conditioning Control	Configuration data/transmission of data
	Zone Temperature Display
	Temperature adjustment characteristics
	Fault handling

Function	Feature
Multipurpose Bus	Transfer data to FAL test system
AMU & FWS	General Functions/Design
ECAM Messages	General/Information status
	Failure Class
Chimes	Decompression chime/Smoke alert
FAP operation	Power Up/CIDS Caution/CIDS Info Row
BITE	DIR faults
	Middle Line and subsequent faults
	Top Line and subsequent faults
	FAP failures
	EPSU failures
	Illumination and Reading Light failures
	FWS indication and highly Critical failures
	FAP & CMS indication
Emergency mode	Essential Power
Reset functions	Director resets
	FAP resets
	Electrical resets

3.2.1 Test Cases

For testing this feature, the described BCEVI process was followed as described in the previous chapters. Initially, all the test documentation is prepared in CATEGA II, as depicted in the following figure:

As it can be seen, all the described Test Objects appear in CATEGA II, in order to represent the succession of events and to accurate describe the steps to be followed in the Test Case. This TC is taken by the Test Executors and the steps are used as guidance during the actual testing. The Test Executors figures out how to perform the steps described in the TC over the real test platform in order to obtain the expected results.

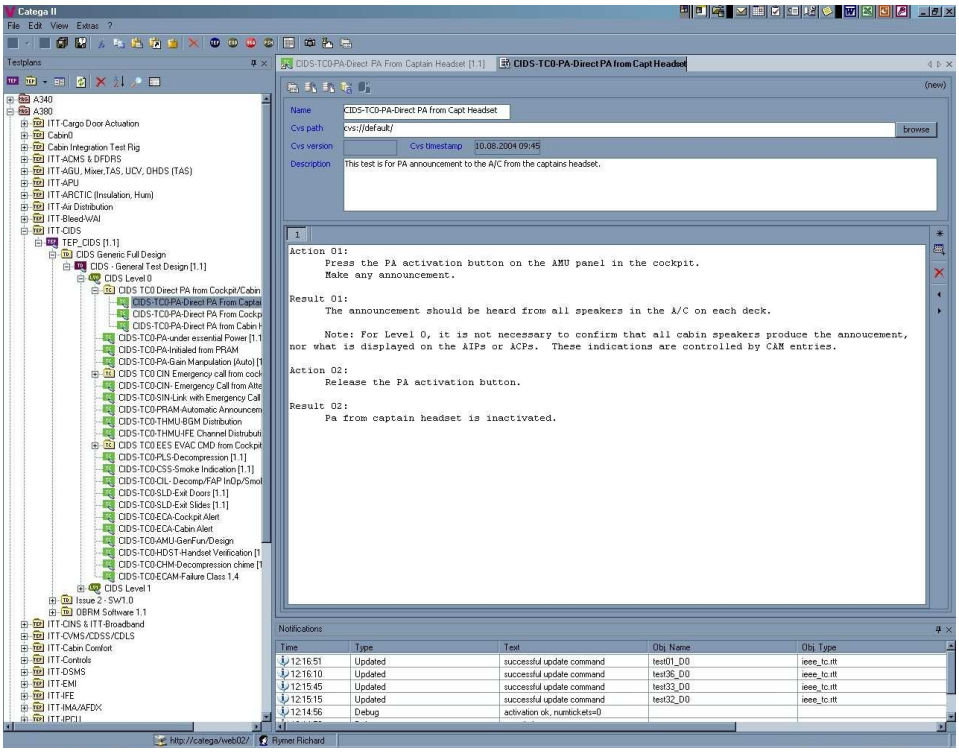


Figure 3-2. Test Case documented in CATEGA II

In order to support the test execution, and to prepare the environment of the test, several scripts are run before the actual test execution. A Python script runs some DEESi commands and sets the values of several CIDS hardware simulators. After this action, the hardware simulators are ready to behave as expected in the TC.

After the initial configuration set by the scripts, several monitor tools are supposed to provide support to the test. These tools would produce the actual results that will be contrasted against the expected results documented in the TC. The monitor tools used by the CIDS team are:

- ADS2: The all-in-one Session should be started. This session contains the required panels to be used during the CIDS testing, regarding I/O information about CIDS discrete signals.

  This session attempts to define all the panels, components and hardware definition to be used both in CIDS and SDF. However, since the stability of the panels is in constant evolution, some problems were found while integrating both systems in one ADS2 session. Therefore, a separate session was designed for each, with the expectation of a future complete integration between the systems.

  An example of the used panels in the TCs for CIDS is shown below. These are related to the Doors & Slides Management System and the Power Control for every system in the DIRs.

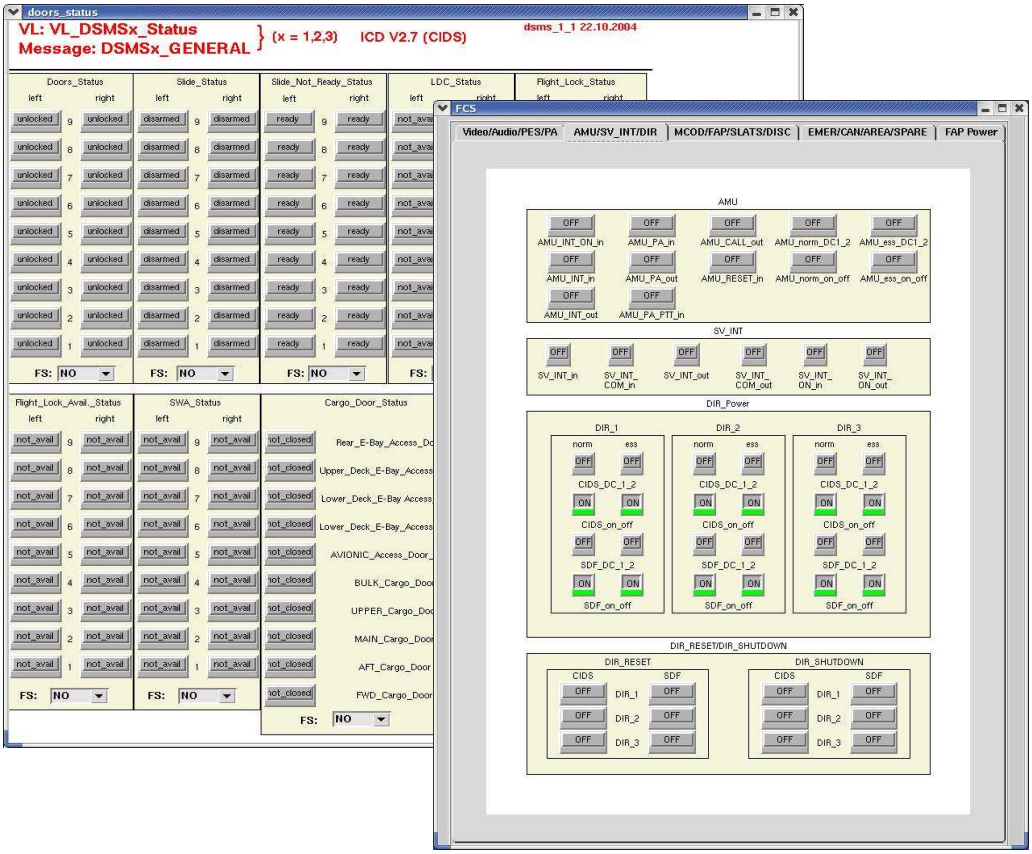


Figure 3-3. DSMS and Power Control panels

- ZOC: This tool is connected to the serial port of one of the DIRs and provides logging information about the current activities and message interchange happening in the director.

- fdXplorer: This tool is used to verify the raw AFDX messages being transmitted in the network. It is useful for a further analysis and comparison of the message structure specified in the system and test requirements.

- CMS Tool: This tool provides an analysis of the avionics system and an interpreter of the Fault Messages generated by the BITE function of CIDS. This is an external production tool and not a part of the simulations.

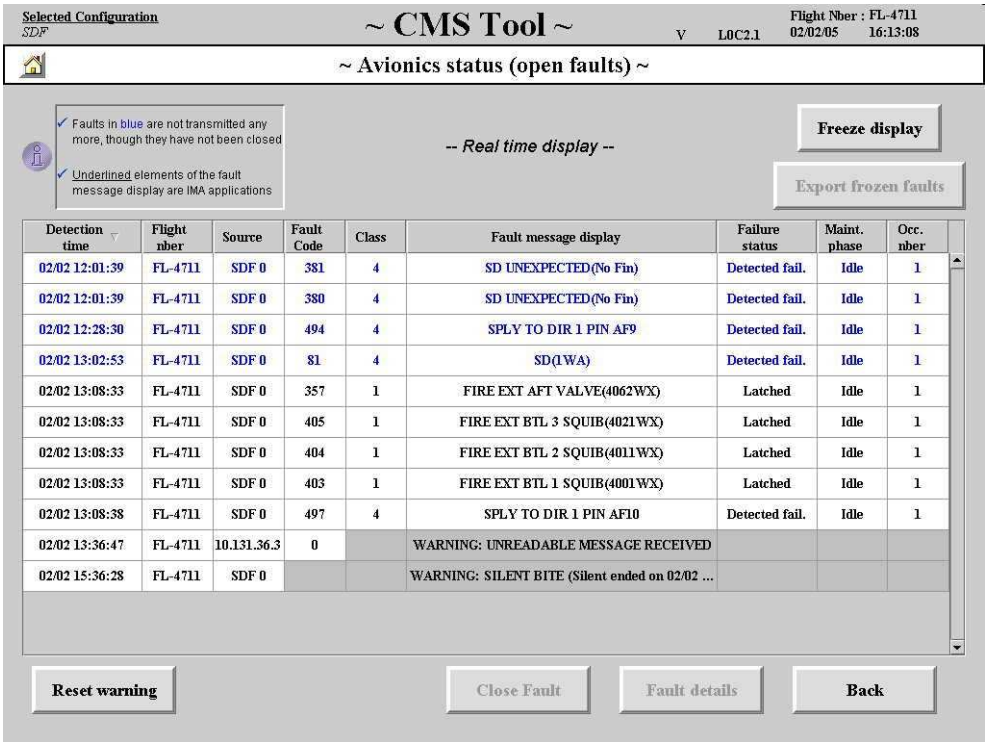


Figure 3-4. CMS Tool

After the configuration of all the tools and the ADS2 session, the Actions in the TC script is followed and the expected results are validated against the results generated by CIDS based on the stimuli provided by the simulators. For this specific Case Study, the verification of the Test Cases for the feature Passenger Address (CIDS-TC0-PA-Direct PA from Cockpit/Cabin) was selected. For this activity, two handsets were used and two PA zones were defined in the SIB; actually, as shown in Figure 3-2. Test Case documented in CATEGA II, a group of TCs for Level 0 represent the minimum set of points of validation according to the system requirements for the Passenger Address feature. Regarding this specific validation, sometimes the expected results of the TCs were not satisfied due to problems with the Original Equipment (in this case, the handsets), with the DEESi simulators, and also of course due to the deviations in the system.

A description of an extraordinary behaviour of the system is annotated in a log book administered by the Test Execution team. This behaviour represents a deviation from the corresponding result described in the TC. When it is manifested, a PR is created in CATEGA II with all the concerning information, as illustrated in the following figure:

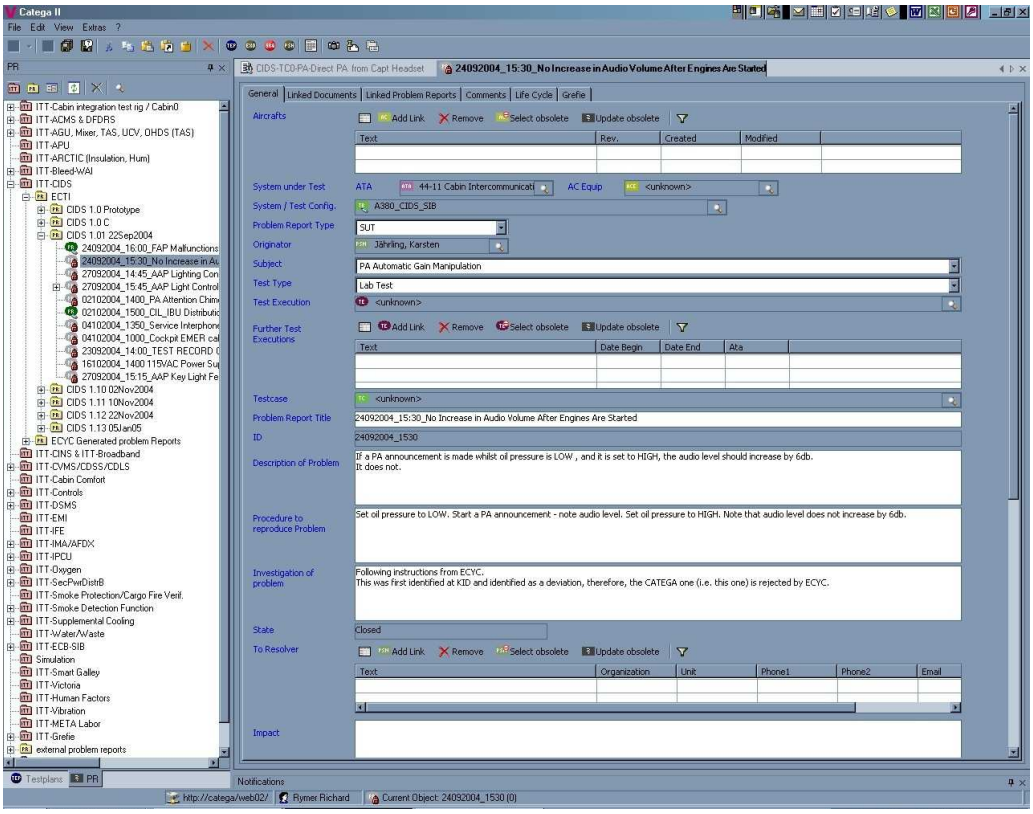


Figure 3-5. Problem Report in CATEGA II

At the end, a Test Report is generated consolidating the results criteria (pass or fail) of every test case, including also the correspondent Test Feature. This report includes metrics about the number of passed, failed and ongoing TCs for a specific CIDS function, descriptive pie charts and a Verification Matrix that includes also the level of the test cases and the general progress of the testing activities.

3.2.2 Multipurpose Bus Analysis

For this specific testing, the process was deviated since it required further investigation, and the execution of this test was requested to be re-scheduled due to management decisions. At the beginning, the Purchaser Technical Specification (PTS) was used. This document contains the Operational Functions of the feature to be tested, in this case, the Multipurpose Bus (MPB). The PTS establishes as requirements that the MPB receives data sent by AFDX and that the update rate should be 1000 ms.

The features to be tested, described also in the PTS are Passenger Address, Service Interphone, Evacuation Signalling, Cabin Interphone, Passenger Lighted Signs, Emergency Crew Alerting System (also known as Hijacking Alert), Sterile Cockpit, Director State (defined by the Sign

State Matrix) and CIDS internal information such as the status of the cockpit door, excessive altitude, etc.

Besides the tools presented in the previous section, another tool that provides support for the testing is Ethereal, where the update rate can be determined after 10 seconds of lecture of messages exchanged over the AFDX network, confirming the expected updating rate from the PTS. This message analysis is also supported by the fdXplorer tool, where the payload information of the AFDX frame is decoded into the message structure. This structure is compared with the format of the output message described also in an appendix of the PTS.

After several tests, some problems were found: the signals were activated, but the DIRs do not send AFDX messages, or they send the messages through another virtual link. In order to confirm if the signals are really being sent by the DIRs, a voltage measurement is carried out in the Connectors Panels in the BOP cabinet. The Connectors Panels represent a matrix of signals generated by every director in the cabinet.

Based on the findings annotated in the testing log, a PR was raised for every signal that exhibit a behaviour different to the one specified in the PTS. The PRs related to this testing are only linked to the specifications, since there were no time for test case definition due to the change of priority and level from the management. The rest of the PR process continued as defined for CIDS.

3.3 Advantages and Disadvantages of Testing Strategy

The testing activities performed in CIDS in order to verify and validate the functions of the system follow the established BCEVI Testing Process. Test Design and Test Execution are completely documented into CATEGA II, following the established communication path. The complete cycle of the PRs is also carried out using CATEGA II, with the access not only for the Test team, but also for the Design team, who have the possibility of verifying and modifying the status of the PR according to the evolution in the solution of the deviation.

Regarding also the Test team, since it is a very complex system, the job distribution inside the team was adequate. The teamwork is demonstrated during the test execution phase, because several panels should be manipulated and also multiple variables should be verified, as well as the specific configuration of every TC. These tasks are performed not only by the Test team, but also with the people in charge of the SIB maintenance and configuration, producing as a reliable result of the testing.

It is important to mention that for a more precise support for the CIDS testing, ADS2 tool panels have a very important role. The constant evolution of the testing has requested new functionality in the panels elaborated with Tcl/Tk that have been successfully adapted for setting values in the CVT. Furthermore, this advantage is not restrictive only for CIDS, but also for SDF.

In order to automate some of the environment configuration steps, several scripts were developed. Using Python and Tcl/Tk, these scripts helped to reduce the time invested in the initialisation of the SIB in order to run the necessary TCs. Since some of the stimuli for the simulators are provided through DEESi, a graphical interface was also developed to facilitate the interaction with the simulators.

As a final advantage, the use of the CMS Tool, an actual production software tool, is very useful in the context of the SIB environment. This piece of software is not a simulator, but a real tool that is attached to the CIDS in the final operation. Testing with this tool instead of using a simulator represent an opportunity for directly validate the answers of the SUT in a real environment.

Difficulties appeared also in the course of the CIDS test strategy. One of these disadvantages is the fact that the Design team is reluctant to use CATEGA II. Despite they have access to the tool, the PRs follow-up is not performed in time and form as described in the process. This represents delays in the resolution of deviations and affects the testing schedule.

As presented in the section 3.2.2, due to demands of the Design team, some testing is performed directly comparing the system specifications without any documented TC supporting the testing activity. This process deviation, even under time constraints, may impact in the quality of the results and produce inexact status results.

A severe problem found during the comparison of the expected against the actual results was the inaccuracy of the ADS panels to reflect the CVT values. Some components report their status in different voltage ranges (different values of logic 0 and 1 depending of the voltage value), but the interpretation of the devRTcore is not correctly since it keeps a standard range for all the signal voltage. This resulted in the direct measurement of the signal voltage in the Connectors Panels in order to avoid the incorrect display of the values of the discrete in the ADS2 panels.

Since CIDS is a large and complex system with several interfaces, a software update constituted by the release of a new version represents a re-execution of the TCs for levels 0 and 1 in order to verify that the PRs from the previous version are closed. The frequency of these updates is high, and not only about the CIDS software, but also with the SIB hardware simulators and original equipment. Even it is necessary in the natural evolution of the testing, the impact in schedule is high.

Finally, a dependency of SDF was found recently during the CIDS testing. It is necessary to initialise and pre-configure the environment of SDF in order to execute some specific TCs for CIDS. However, the CIDS Test Execution team has no complete information about the functionality of SDF, producing then incorrect suppositions about the possible source of problems during the test execution.

3.4 Suggestions for Improvement

After analysing the opportunity areas found in this case study, the suggestions below are formulated in order to improve the overall performance of the current test activities for CIDS and optimise the results of the work products.

- Software simulators (stubs) for AFDX messaging interchange between CIDS and the hardware simulation devices not yet delivered by the supplier. This stubs would be useful is some testing is to be executed within the planned schedule but the resources are not available at the moment. An example of this solution in the industry can be found in [12]. This would help to avoid delay, like in the case of the PISA simulators that were delivered in an advanced stage of the testing and due to some failures, returned to the supplier, impacting in the testing schedule.

- Establish a Configuration and Change Management process in place. This process should be focused in the documented control of the assets, understood as hardware simulator, software version (including scripts and path location) and document version. In several occasions, the test preparation and execution are disrupted because some assets were modified without previous notice to the Test team, or the version of a given specification was modified. The Configuration and Change Management process reduces the risk of rework due to use of out-of-date assets and guarantees the awareness of all the Test team if one modification is done. This suggestion would apply also for SDF.

- Generate accurate statistics of test execution, test preparation and test reporting times, as well as the time expended in follow-up of PRs. Fill in CATEGA II this metrics that will help to provide better estimates for planning and scheduling for the test design and execution activities. This suggestion would apply also for SDF.

- Regarding the PRs follow-up, a CATEGA II weekly report containing outstanding and high priority PRs would be sent to the Design team. The objective of this report is to make the Design team aware of the necessities of the Test team and motivate the use of the tool to update and analyse the relevant PRs.

- The TCs can be defined in such a flexible manner that reflects the steps to be followed and the tools to be used in the SIB. Not only a generic description taken from the system requirements should be provided, but also specific scenarios that can be modified and controlled by a change mechanism (in CATEGA II or in an additional tool). As suggested in [6], the test procedure should be as detailed as possible, covering all the possible alternatives and providing strict steps in order to ensure the consistency of the expected results and covering all the possible range of outputs.

These suggestions were submitted to the Test Execution team in the BCEVI1 department, in charge of the V&V of CIDS. It will be discussed internally which of those suggestions can be selected for a further implementation in the test campaign for the current software of the A380 aircraft or in future test campaigns for new and upcoming system upgrades for other Airbus models.

Some of the suggestions formulated for the test strategy of CIDS depend on modifications to the testing process (theoretical component) and can be proposed with the support of current tendencies in the documented in the concerning literature about testing strategies from other Independent Verification & Validation sources. The other proposals in reference to the technical component can be prepared using the available tools and languages, but allocating resources and scheduling time for the implementation of the solution.

The next chapter will cover in a similar sense the description of the system, as well as the testing strategies for the Smoke Detection Function. One important difference between CIDS and SDF is the size and complexity of the system, because SDF is significantly smaller and less complex than CIDS. As it will be seen, this difference represents a repercussion in the test strategy, size of the test team and test platform configuration. This second industrial testing instance looks for providing another point of view about the V&V activities in the same testing sector of Airbus.

4. Case Study: Smoke Detection Function

The Case Study exposed in this chapter provides an overview about a special functionality of the Core Cabin Systems of the A380 aircraft: the Smoke Detection Function. SDF controls a critical feature of the aircraft related to the security of the aircraft in any risk of fire in the Cargo and Avionics sections. The requirements for this system are very restrictive and specialised for the smoke function. However in this occasion the validation and verification request is satisfied using a different methodology that will be explained in the following sections of this chapter.

The brief description of the system is followed by the two industrial testing instances that will be analysed, as done in the Case Study of CIDS. One more time, improvement suggestions will be provided based in the proposals found in the referenced documents. In this manner, the goal of providing real examples as base of process improvement is attained from another point of view.

4.1 System Description

This system implements the ATA chapter 26. This chapter is related to the Fire Detection and Extinguishing, and documented in [1]. In the referent to Pressurised Areas, the cargo compartments, avionics ventilation system, lavatories and crew rest compartments have a smoke detection protection installed. The cargo compartments have also extinguishing bottles installed.

Lavatory smoke detection is given by ambient smoke detectors installed in the lavatories ceiling (one detector per lavatory). Each lavatory waste bin has an automatic fire extinguishing system. These sub-compartments have sufficient fire/smoke detection: trolley lift, flight crew rest compartments, cabin crew rest compartments, and passenger rest compartments. Portable extinguishers are installed in the cockpit and in the passenger compartments. However, both features were not part of the scope of this testing.

The Avionics Bay Smoke Detection consists on a dual-loop smoke detection system that is installed in the main avionics bay left hand and right hand side, emergency avionics bay, aft avionics bay and In-Flight Entertainment Centre. Dual optical smoke detectors are installed in the air extraction ducts. The detectors are connected to ECAM added to the On board Maintenance System (OMS). The Cargo Compartment Fire detection is implemented by a dual-loop smoke detection system that is installed with dual optical smoke detectors installed in the ceiling, each pair in a cavity of the ceiling. The system operates a combined AND/OR logic in the CIDS which is connected to the ECAM and OMS. The following diagram illustrates the Smoke Detection System Architecture:

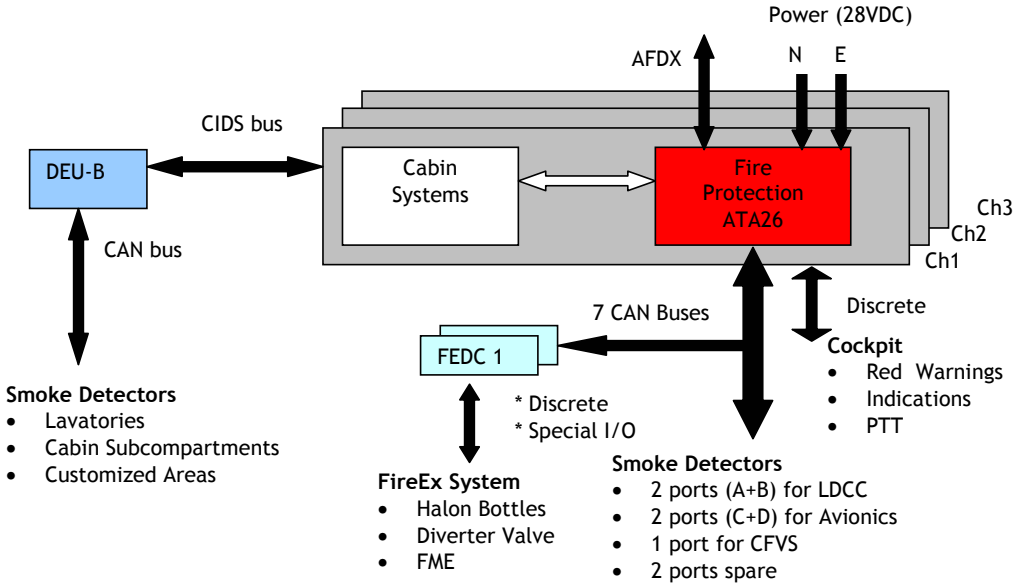


Figure 4-1. Smoke Detection Function (SDF) System Architecture

Regarding Fire Suppression, three fire suppression bottles are installed in the FWD cargo compartment. The contents can be discharged into the forward or aft (including bulk) cargo compartment. The fire extinguishing system is manually started by the flight crew when the Fire/Smoke Detection System gives a warning to the cockpit. By operating the push button switch FWD AGENT the A- and B- squibs of the FWD cartridges in the three fire extinguishing bottles will be fired simultaneously. By operating the pushbutton switch AFT AGENT the A- and B- squibs of the AFT cartridges in the three fire extinguishing bottles will be fired simultaneously.

When a fire detection is confirmed by the FDU (Fire Detection Unit), these fire warnings are given in the cockpit on: the CARGO SMOKE control panel: ICP-9, the Engine Warning Display, the System Display, and the aural warning sounds. The pushbutton indicators FWD AGENT and AFT AGENT show a red coloured legend SMOKE if there is detected smoke in FWD respectively AFT/BULK cargo compartment. The indicator DISCH show a white coloured legend BTL 1, BTL 2, or BTL 3 when detected low pressure in one of the three fire extinguishing bottles. By operating the pushbutton switch TEST, the smoke detection control function is receiving signals about simulated low pressure in the three fire extinguishing bottles and test active, and is activating the squib circuit continuity test. If one squib per outlet is okay the indicator with the white coloured legend SQUIB (FWD or AFT AGENT) comes on.

4.2 Current Testing Scenario

The testing activity of SDF was carried on by a team of two persons, working in the morning with the TDS and PR follow up and in the afternoon executing the tests in the laboratory. This section describes two examples of how these tests were designed and executed in the scope of the BCEVI test process, and also using the tools described in the previous chapters.

4.2.1 Test Data Sheet

The Test Data Sheet (TDS) is a document used by Airbus that defines a specific set of Test Inputs applicable for the testing of specific requirement of the system. It establishes the Requirements Reference (Specification, Traceability Matrix, etc.), Initial Conditions, Data Recording, Testing Type (Simulation or Test), Level (Aircraft, System and Component), Simulation and Test Tools, as part of the general setup of the testing. For SDF, the TDS is prepared by the Design team located in the Airbus facilities in Bremen and submitted to the Testing team to be executed.

The TDS includes a description of the feature to be tested. It provides a complete overview of the feature, including some diagrams to clarify the behaviour of the feature during the testing and to help the tester to understand the expected results. If any previous TDS should be ran before the current, it is stated during the description. After the explanation of the feature, a general outline of the test inputs to be performed is given.

The Test Inputs (TIP) appear after the description, in a section named Product Requirements. This section is divided into parts; each of one is focused in a specific set of system requirements to be tested. Every TIP contains an optional list of involved System Requirements, a series of steps to be performed, and an Expected Reaction.

For this specific Case Study, the TDS.1.2.2_AVIONICS_FAILED was selected as an example of the development and execution of a system black box testing for level 1 of the SDF. This TDS covers the FAILED signalling for smoke detectors. At the same time “single detection mode” is going to be validated. The single detection mode works setting one S/D of a ceiling cavity into a FAILED state, will discard the failed S/D out of the current configuration table, and the remaining S/D is declared to “single detection mode”. If in this single detection mode the S/D changes the state to ALARM, then a compartment ALARM will be set to TRUE. The second part of this TDS covers FAILED signalling for a smoke detector that send wrong content on the CAN bus. In particular a sensor will be declared FAILED if it reports itself as both in STANDBY and ALARM or in none of these states. In the third part, the contamination mode is verified. Then the combination of two FAILED S/D is covered in the fourth part and the combination of one FAILED and one OFF detector is verified in the fifth part.

In order to prepare the configuration for the testing, a DEESi script had to be run. This script sets all the simulators from Active mode (default) to Off mode, and then to OE mode. Therefore, the hardware simulators are ready to perform as the original equipment and in this manner the input can be manipulated directly according to the TIP requirements. Also, the a380-sdf-Banz2 Session should be started in the ADS2 tool. This session contains the required panels to be used during the SDF testing, regarding I/O information about SDF discrete signals, AFDX bits for the FWS, and CMS messages.

It is also important to configure the monitor of the CAN before performing the execution of the test input. CANalyzer is the tool used to monitor and verify that the corresponding messages are sent to the smoke detectors and answered by them through the preconfigured CAN.

In the TDS showed in this example, the hardware smoke detector simulators are set by pairs in the configuration of the CAN bus, i.e., smoke detectors of CAN-A and CAN-B are disposed in parallel, as well as the smoke detectors of CAN-C and CAN-D according the following diagram:

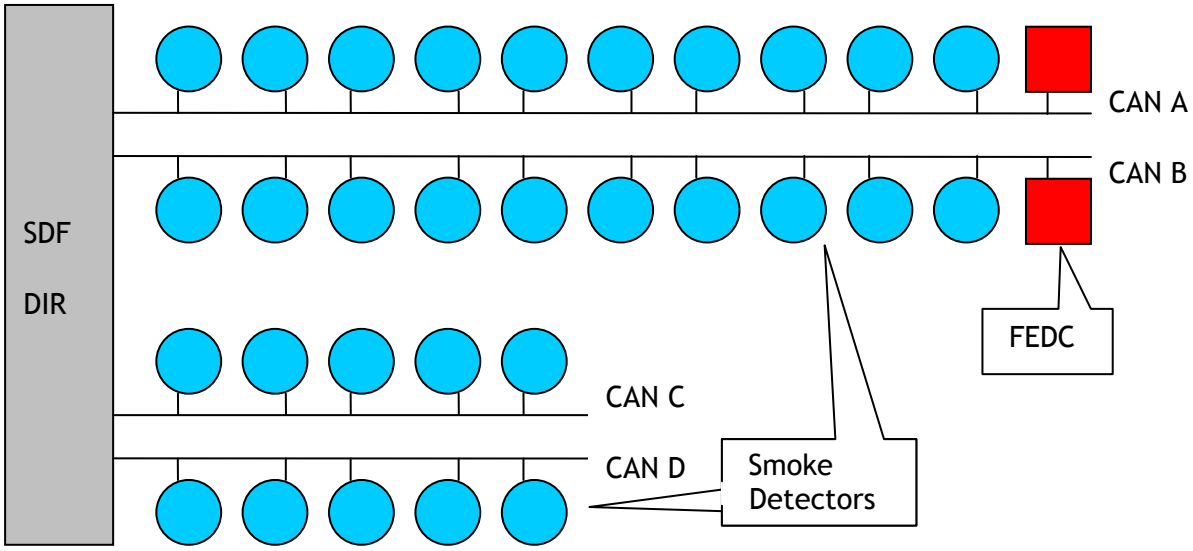


Figure 4-2. Configuration of the CAN buses by pairs

Every Test Input was executed as described in the TDS, using the support monitor toolsin order to verify that the expected results are generated. In this specific test the following tools were used:

- ADS2: The CMS BITE monitor, a panel included in this tool, allows the visualisation of the failure messages from the three DIRs and the content of these messages. Some messages,

identified by the Failure Message Code and the Failure Message Description, are expected during this test. The FM Code was described in the Expected Result and its transmission is confirmed using this panel. Expected FM Codes include indications for FAIL signal of one of the elements of the pair of smoke detectors, ALARM signal and ERROR of the pair of detectors.

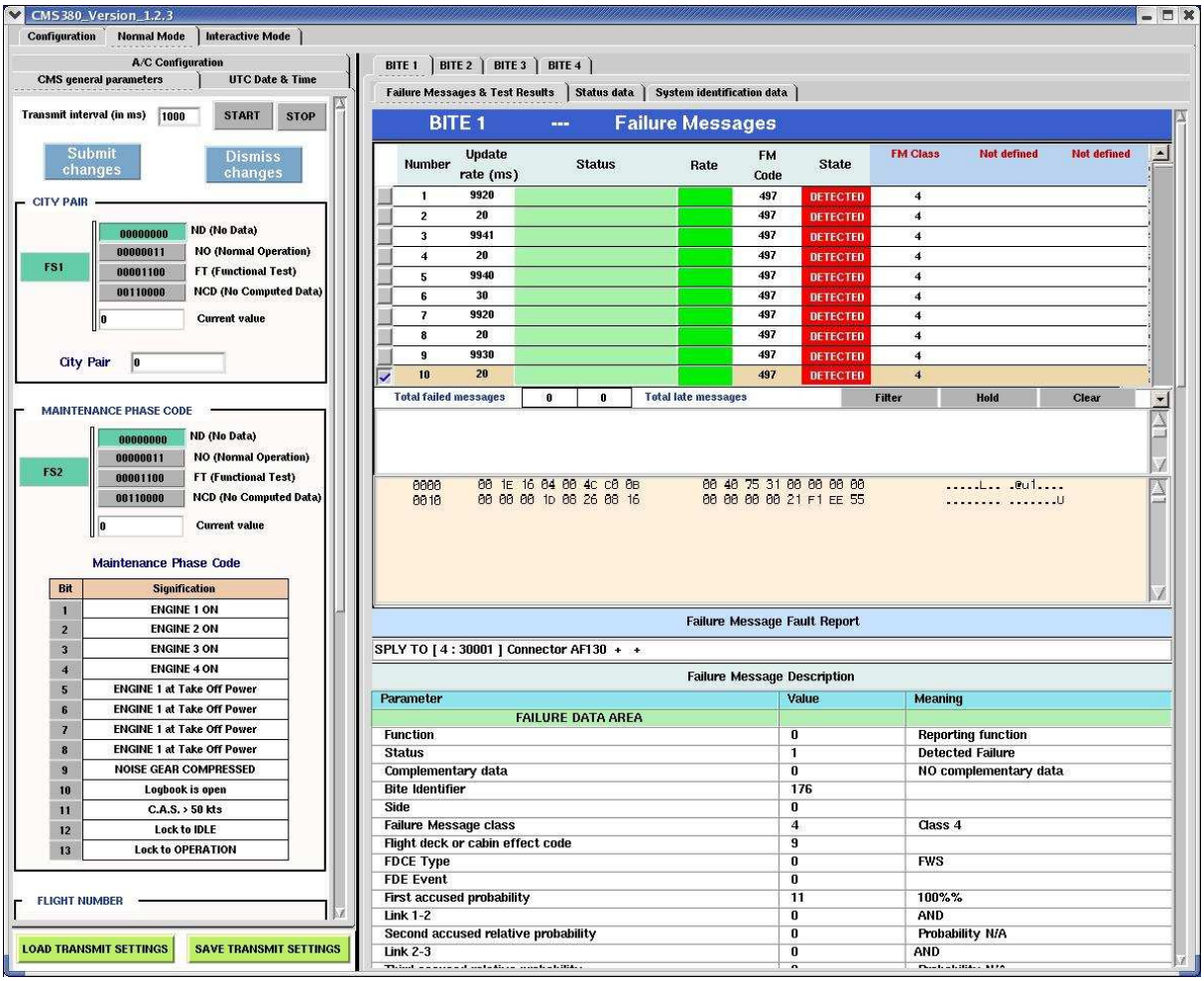


Figure 4-3. CMS BITE Message monitor panel

There are also some panels that allow the validation of the discretes and the bits of the AFDX that are set for the FWS. The expected reaction included in the TDS describes this information and the tester verifies that such description agrees with the displayed information in the ADS2 panels.

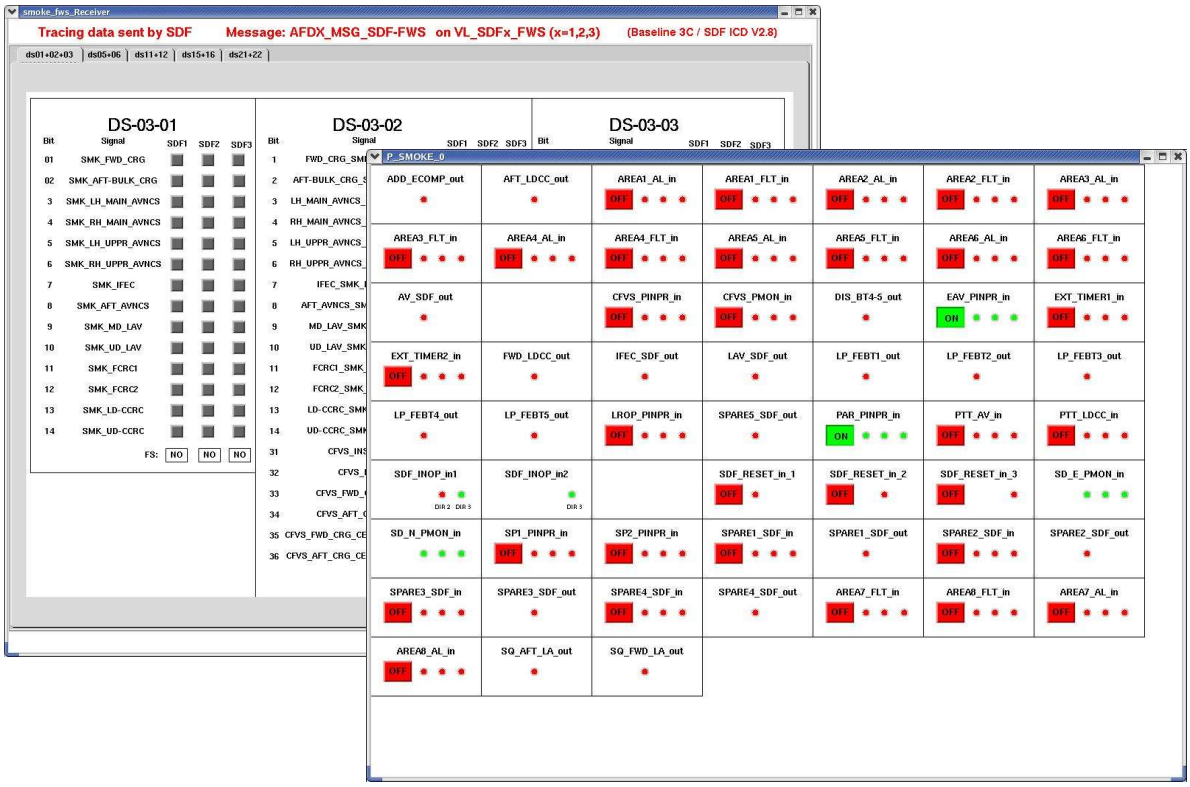


Figure 4-4. AFDX and Smoke Discrete panels

- ZOC: using this tool, the MMC code expected as described in the TDS is confirmed. This MMC code is associated to the FM code verified in the CMS BITE Monitor panel of ADS2.

- CANalyzer: there is a protocol that is initiated into the S/D CAN every time a DIR is power on. This protocol includes the transmission of the device context parameters (date, time, flight number, etc) and a cyclic polling of the smoke detectors start. After the correct initialisation of the S/Ds, some messages are expected to be sent or the polling of a failed S/D to be stopped; all of this behaviour is defined in the TDS and also validated using the CAN Trace windows in CANalyzer. In the following figure, the trace of every CAN bus is shown while polling.

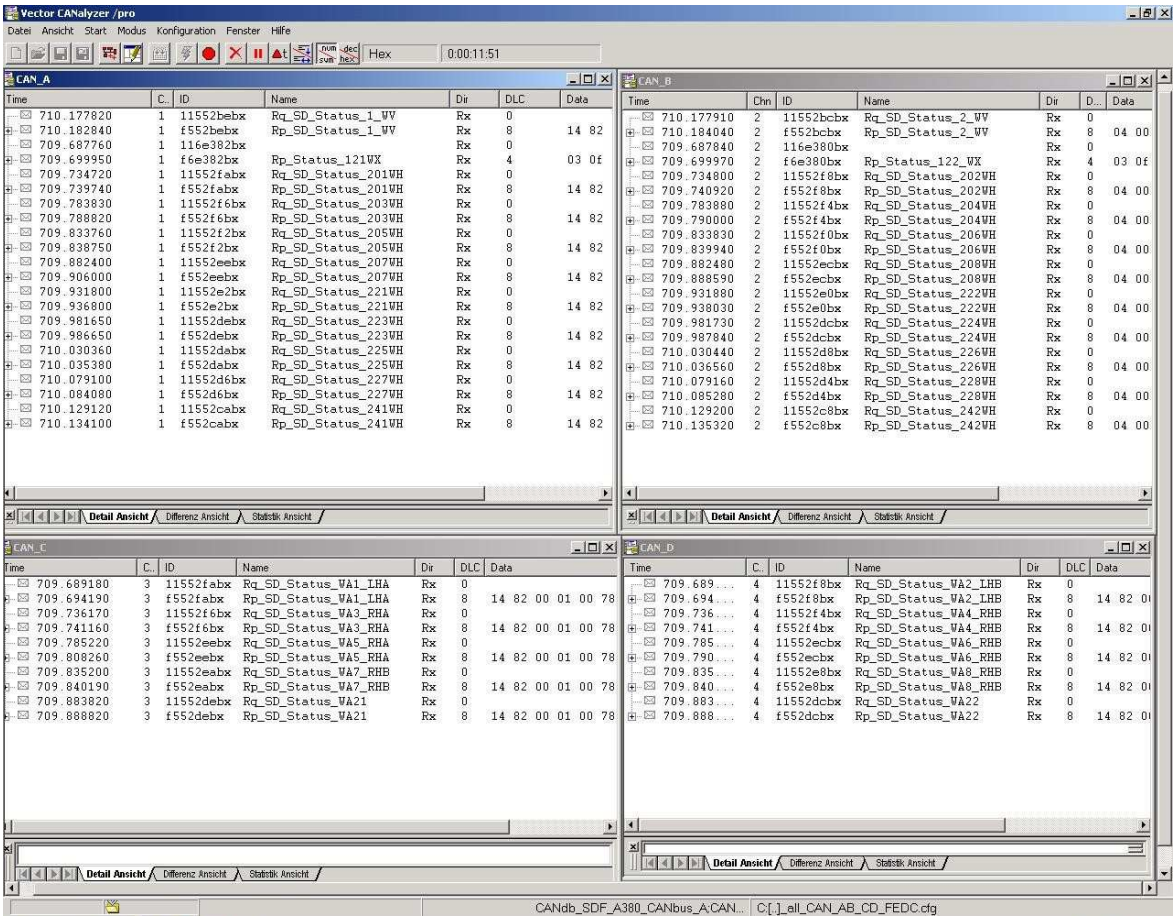


Figure 4-5. CANalyzer trace

- fdXplorer: This tool is used to verify that the raw AFDX messages are correctly sent, providing a second source of analysis for the validation.

Once the validation of the expected results was performed, the appropriate criterion is registered in the corresponding log of the TDS. There is a log for every TDS to be tested. In this log, the field “TIP” is completed with the corresponding TIP being tested, and in the “SW Version” field the current software version being tested is written. The rest of the fields should be completed with the values “OK” or “NOK” (Not OK), depending on the result of the comparison of the validation of the expected results against the real result. An example of this log is shown in the following figure.

TIP	CAN bus reaction	Status	AFDX Transmission	Discrete	Maintenance Message / via RS232	Remark	overall result	SW Version
43	OK	OK	NOK	NOK	NA	Error in discrete	NOK	V7
44	OK	OK	OK	OK	OK		OK	V7

Figure 4-6. SDF test log

After the result is recorded in the log, for every TIP whose overall result is NOK a PR is raised in CATEGA II. The PR is managed the same way as in CIDS.

There are some special conditions applicable to execute again the test: when the result of the test is not constant, or there is a mistake in the configuration, or if the panels discrete do not reflect the expected result in a constant way or there was a failure in the lecture and/or execution of the test.

However, it is possible that after several attempts for completing the testing, or because the ADS2 panels do not reflect the right discrete indication, or due to a petition from the Design team, the Connectors Panels in the BOP cabinet are used to verify the output of the signals along with the voltmeters. These voltage measurements are contrasted with the original configuration of the SIB and with the system requirements specifications for SDF. In most of the cases during this project work, the voltage measurements were right and the panel reflected inaccurate indications.

4.2.2 System Start-up Analysis

Not only from a Test Data Sheet, but also testing can be formulated in order to investigate the real source of the deviations appearing continuously from the expected results from TDSs. In this case, the following test was designed and executed in order to find a dependency of the position of the director in the trays of the UUT cabinet, or to find if one or more of the directors produced unexpected messages.

The test was formulated by defining a combination of positions and status of the DIRs during the system start-up, and then visualizing the output BITE messages using the CMS BITE Monitor panel from ADS2. The DIRs are identified by their Serial Number, the positions by the tray number (T1, T2 or T3), and the messages are visualised in the CMS BITE Monitor panel identified by the virtual link whom they are connected. Four different combinations of the arrangement of the DIRs were proposed, and a visualisation of the messages for a period of 60 seconds was performed. It is important to mention that the preconditions of this testing guarantee that no error messages are expected during start-up. The results were logged into a comparison table in order to identify the different behaviours or the DIRs during the proposed combination:

DIR Ser. No. in T1	DIR Ser. No. in T2	DIR Ser. No. in T3	VL_SDF1_SERVICE Messages	VL_SDF2_SERVICE Messages	VL_SDF3_SERVICE Messages
15/ON	25/OFF	30/OFF	only GH periodical	NM	NM
15/OFF	25/ON	30/OFF	NM	1. No GH 2. 403, 404, 405, 357, ("LATCHED" state) 464,467,517,508,511 (periodical)	NM
15/OFF	25/OFF	30/ON	NM	NM	only GH periodical
30/ON	15/OFF	25/OFF	only GH periodical	NM	NM
30/OFF	15/ON	25/OFF	NM	1. 3x2 GH 2. 464,467,517,508, 511 (periodical)	NM
30/OFF	15/OFF	25/ON	NM	NM	1. No GH 2. 403, 404, 405, 357 ("LATCHED" state)
25/ON	30/OFF	15/OFF	1. No GH 2. 403, 404, 405, 357 ("LATCHED" state)	NM	NM
25/OFF	30/ON	15/OFF	NM	1. 2x2 GH 2. 464,467,517,508, 511 (periodical)	NM
25/OFF	30/OFF	15/ON	NM	NM	only GH periodical
15/ON	30/OFF	25/OFF	only GH periodical	NM	NM
15/OFF	30/ON	25/OFF	NM	1. 3x2 GH 2. 464,467,517,508, 511 (periodical)	NM
15/OFF	30/OFF	25/ON	NM	NM	1. No GH 2. 403, 404, 405, 357 ("LATCHED" state)

Special messages:

* NM = No Messages

* GH = Good Health Message

For a better analysis of these results, it is necessary to keep in mind that when a DIR is in OFF mode, no messages are sent at all. It is important also to notice that when a DIR is in ON mode, the “Good Health” Message is expected to be sent periodically, meaning that the SDF card have no

problems found by the BITE system. Therefore, the expected results were to receive only “Good Health” Messages from the director that is in ON mode, in any combination of trays and serial numbers, meanwhile the directors in OFF mode should not send any message.

The results of this test are not actually documented in a log or in CATEGA II, but they served only for a research purpose about the dependency of the directors on their positions on the tray. As it can be seen in the table above and compared to the expected results, it can be noticed that the director with the serial number 25, when in the ON mode, sends the messages with FM Code 403, 404, 405 and 357 ("LATCHED" state) in each position it had. On the other hand, regardless what director is located in tray 2, it sends the messages with FM Code 464, 467, 517, 508 and 511, and none GH Message.

This testing is not conclusive, since a black box technique was used and the actual sources of discrepancies between the expected results and the actual results cannot be definitively established. The results were commented with the Design team in order to make them aware of the potential problems this behaviour can provoke in current and future TDS.

4.3 Advantages and Disadvantages of Testing Strategy

The testing strategy followed to validate and verify SDF offers a different perspective of the established testing process in Airbus. It follows the established procedure and uses the appropriate tools to configure and verify the testing requested.

An effective approach to corroborate that SDF is correctly setting the discrete signals is, as seen in section 4.2.1, the measurement of the actual voltage that reflects the behaviour of the Directors. This action aids to eliminate any possible inaccuracy in the comparison of the expected result (from TDS) against the current result (as displayed incorrectly in the ADS2 panels, due to an error in the tool) and to confirm or reject the existence of a deviation in the specific test.

Another advantage in the testing strategy of SDF is the use of CANalyzer, and specifically the functionality of CAPL. The CAN Access Programming Language allows to program CANalyzer for emulate and modify the content of some messages and for analyze the content of the payload of a given message. The flexibility of this tool helps to generate a vast test input data and verifies the behaviour of the CAN under extreme situations.

If there is a question about the correct interpretation of the input data or an execution step in any TIP from a given TDS, the SDF test team contacts the Design team who is in charge of the preparation of such TDS. They interchange impressions about the correct meaning of the instructions

of a determined TIP, or decide if the TDS contains a typographic or conceptual mistake. This situation helps to improve the test execution and to avoid misunderstanding that may lead to the generation of incorrect problem reports.

A remarkable point in this test strategy is the self motivation to develop new internal test cases, for example measuring the voltage of the signal output, whose purpose is to contribute to the clarification of possible points of deviation for FEDC testing and for research purposes. The additional research performed by the Test team provides elements to support the understanding of the behaviour of the system and feedback to the Design team in order to formulate more precise TDS.

A final advantage is the relative independence of SDF from CIDS. Even both systems reside in the same Director, SDF can run with no dependency from CIDS, due to SDF is loaded in a separate card inside the Director. Regarding the test strategy, the SDF test execution can be performed autonomously saving time and reducing resource consumption.

However, some of the activities related to this strategy present opportunity areas. One of them is the follow-up of PRs in CATEGA II. When the Test team registers a PR in CATEGA II, the normal process establishes that the Design team should log into CATEGA II and they themselves accept or reject the PR using the same tool. Nevertheless, the Design team has no access to use CATEGA II. Therefore, the Test team generates a CATEGA II Report in Microsoft Word, extracting the information about the real PR and then sent the Word report by email to the Design team. This PR management implies duplicity of efforts and a deviation of the process.

Since the Design team generates the TDSs for the Test team, not all of the documents are available when they are needed. There is a strong dependency of the release of the TDSs from a separate entity (Design team). This situation impacts on delays that are not really manageable by the Team test.

A final disadvantage is that the simulators and the original equipment for the FEDC subsystem still exhibit unconformities that lead to unexpected results. After a considerable period of time, the FEDC devices suddenly set the error bit, send warning messages, becoming the system inoperable, guiding to a re-initialisation of the FEDC and S/D simulators.

4.4 Suggestions for Improvement

Based on the opportunity areas found in this case study, the following suggestions were originated in order to provide support to improve the performance of the current testing activities and optimise the results of the work products.

- A simulation of the S/D in the CAN is suitable to improve research of test cases. Taking advantage of CAPL capabilities in CANalyzer and providing an adequate GUI for the test executor, this simulation can provide alternative scenarios and modified content of messages that can be tested even in the absence of hardware simulators, increasing flexibility of testing. In this way, it is possible to create and send CAN messages, program scenarios, and investigate unexpected system behaviour.

- Unification of the activities of the Test Design Phase mainly focused in the generation of Test Cases for the respective Test Level for SDF. This action will reduce the waiting time for the TDS that in this moment occurs caused by the preparation of the TDS done by the Design team, and it is already performed by the CIDS Test team.

- In order to improve the quality of the software, it is recommended to apply the same criteria for the administration of the Problem Reports as in CIDS. This would help to avoid the duplicity of efforts of the Design and Testing teams, to improve the follow-up of the PR, and to homogenise the activities of the same Test team (BCEVI) for both systems, CIDS and SDF, even if the Design teams are different for each system.

- As mentioned in the SEI CMMI model [3], the Project Tracking and Oversight Process Area recommends that the communication of the summary of the activities should be done at all levels. For this reason, it is recommended to submit the weekly report prepared by the Test team leader not only to the management level, but also to the team members. In this way, the people that work together in the SIB are aware of the work happening around them and can provide help or raise issues about the general activities.

As in the previous Case Study, these suggestions were also submitted to the SDF Test team in the BCEVI1 department. These suggestions would impact not only to the test team, but also to the design team that is part of the testing preparation and works for another sector of Airbus. As presented in this chapter, it can be appreciated that even if the SDF system is tested in the same department as CIDS, the specific features of the smoke detection software create the necessity of a special testing process and of additional configuration of the test platform.

After the analysis of both testing strategies has been performed in the Case Study structure, a final summary with lessons learned, directions for future studies in the field of validation and verification and open problems will be presented next. It will be also concluded if the objectives established at the beginning of this project work were achieved.

5. Summary

In this project work factual instances of independent verification and validation in the industry have been described, and also improvement suggestions have been generated for these activities based on theoretical methods and practical testing techniques.

The results generated in this project, i.e. the improvement suggestions, were requested by the BCEVI1 department in Airbus to be implemented in a future test project and test campaign. The bases for an improved upcoming process are provided with the formulated suggestions during this project.

The current test strategy for the cabin core systems performed in Airbus correspond with some of the proposed theoretical testing techniques presented in chapter 2 of this document. The Design-Based Functional Testing technique is applied both in Test Data Sheets and Test Cases in CATEGA II, since both approaches cover the system requirements in a comprehensive manner. On the other hand, the Cause-Effect Graphing technique can be observed in the definition of the Test Cases for CIDS testing, since the flexibility in the test description looks to wrap the most common input data producing diverse output results. In this way, several System Requirements are tested with one Test Case.

From the perspective of Functional Testing, Negative Testing is applied while defining special cases for informally testing the directors' switchover behaviour both in CIDS and in SDF. In addition, TDSs including error management to test SDF fault tolerance and BITE messages are an example of Negative Testing as well. This kind of testing, as it was exposed in section "Current Testing Scenario" of the Case Studies, represent a core competence of the Testing team since provide more possibilities to discover deviations in the software performance.

The technical suggestions of improvement proposed in this paper potentially provide a short return of investment. Comparing the expended resources in the current testing strategy with the software stubs for CIDS and CAN simulators for SDF, savings appear in time and hardware resources. There would be reduction in the delay time of testing execution due to the hardware simulators, and even more, these simulators can be omitted at all in testing under specific conditions.

One significant point to remark during this project is the importance of the simulation tools during the execution of the tests. The testing environment simulation platform is an essential component for the testing of real-time systems, as CIDS and SDF. In this respect, ADS2 as main motor of the complete simulation of both systems provides a simulation world [16], with conceptual and simulation models that provide the most approximate representation of the additional stimuli received by the CIDS. This simulation is performed within a local scope, before a complete

implementation of the interaction of the cabin systems with the other aircraft systems in the Common Integration Test Rig.

However it is possible also to find problems in the validity of the tool as seen in the display of the discrete signals. Although all the previous validation and verification techniques would be performed on the simulation tool, the modifications cannot be performed immediately because ADS2 is third-party software. This problem remains open until a solution can be implemented by the tool supplier (Tech S.A.T.) or if any other different supplier can support the simulation required for the cabin systems testing.

Other open problems that remain after this project work are relative to costs. Given that a considerable investment was spent during the building of the SIB, the test platform is expected to be reused for further cabin system integration testing. The reuse of the hardware simulators and software tools has not been clearly stated yet, but the aim of this action is to maximise the utilisation of the material already installed in the laboratory facilities.

Regarding delimitation of responsibilities, this problem is delegated to Airbus Cabin & Cargo Customisation Department. A new distribution of the testing activities should be rearranged between the personnel of the department of Validation & Verification of Core Cabin Systems Integration and the personnel of the department of Design of Core Cabin Systems. This new team division was suggested in order to improve the communication of the teams and to facilitate the performance of the testing.

As collected in [17], several experts share the opinion that in the industrial domain, the two major V&V issues are obtaining support in the simulation studies and having analysts who have sufficient knowledge about these activities. On the other side, academic people see the need for research in a broad sense. From my particular point of view, it was quite complicated to find innovative techniques and methodologies related to black box testing in the literature, since much of them were adaptation of already existing techniques. However, with current challenges in the validation and verification areas of industrial software system like in Airbus, further research and preparation of skilled people is expected to be continuously generated to solve the issues previously mentioned.

6. References

[1] Airbus Training Center Hamburg. **General Familiarization Course for A380-800.** 2004.

[2] Balci, Osman. **Verification, Validation, and Certification of Modeling and Simulation Applications** on Proceedings of the 2003 Winter Simulation Conference.

[3] Conwell, Candace L., Enright, Rosemary and Stutzman, Marcia A. **Capability Maturity Models Support of Modeling and Simulation Verification, Validation, and Accreditation** on Proceedings of the 2000 Winter Simulation Conference.

[4] esd GmbH. **Manual of CIDS-DEESi for the SIBs for the Cabin Systems of an A380 Aircraft.** 2004.

[5] Hall, Robert J. and Zisman, Andrea. **Behavioral Models as Service Descriptions** on ICSOC'04, New York, New York, USA. 2004.

[6] Horch, John W. **Practical Guide to Software Quality Management.** 2nd edition. Artech House, 2003

[7] JUnit Group. http://www.junit.org

[8] KID Systeme. **Training Manual for Cabin Intercommunication Data System (CIDS).** 2003.

[9] Lembcke, Ralph. **CIDS DEESi Technologievorhaben CASIV III Projektreview 2002.** Airbus Deutschland, 2002.

[10] Lembcke, Ralph. **Requirements for CIDS DEU – Electrical Environment Simulation System.** Airbus Deutschland, 2004.

[11] Lewis, Robert O. **Independent Verification and Validation: A Life Cycle Engineering Process for Quality Software**. John Wiley & Sons. 1992

[12] Metz, Michael L. and Jordan, Jack. **Verification of Object-Oriented Simulation Designs** on Proceedings of the 2001 Winter Simulation Conference.

[13] Quirk, W.J. **Verification and Validation of Real-Time Software**. Springer-Verlag, 1985.

[14] Richards Adrion, W., Branstad, Martha A. and Cherniavsky, John C. **Validation, Verification, and Testing of Computer Software** in Computing Surveys, Vol. 14, No. 2, June 1982.

[15] Roache, Patrick J. **Verification and Validation in Computational Science and Engineering**. Hermosa Publishers, 1998.

[16] Sargent, Robert G. **Verification and Validation of Simulation Models** on Proceedings of the 2003 Winter Simulation Conference.

[17] Sargent, Robert G.; Glasow, Priscila A.; Kleijnen, Jack P.C.; Law, Averill M.; McGregor, Ian; Youngblood, Simone. **Strategic Directions in Verification, Validation, and Accreditation Research** on Proceedings of the 2000 Winter Simulation Conference.

[18] Schulmeyer, Gordon G. and McManus, James I. **Handbook of Software Quality Assurance**. 3rd Edition. Prentice Hall, 1998.

[19] Tech S.A.T. GmbH. **CMS380 Simulation User Manual**. 2004

[20] Yang, Guang; Sangiovanni-Vicentelli, Alberto; Watanabe, Yosinori; and Balarin Felice. **Separation of Concerns: Overhead in Modeling and Efficient Simulation Techniques** on EMSOFT'04, Pisa, Italy.