A Data Minig Framework for Management of Ground Testing of the A400M Aircraft

Biscarri, F.[*], Monedero, I., Larios, D.F., Barbancho, J.

Abstract

A hundred development and certification ground tests must be performed before the A400M aircraft's first flight. Ground testing is an essential but also expensive process performed by the Airbus Defense and Space Company (Airbus DS) in the manufacturing cycle of an aircraft. All parts of the aircraft, as well as systems, must be meticulously checked in this type of test; the process involves a large number of company staff and a significant amount of resources. Ground testing involves a replay of a large group of tests due to incidences (test faults) in testing. One or more incidences in a test imply that the test will be repeated, which requires a significant investment of resources and time by the company's engineers. An innovative decision support framework to manage the ground testing sequence is presented. The framework described in this work is the result of a project of the Electronic Technology Department at the University of Seville (Spain) and Airbus DS. This software compares several different test sequences from several aircraft. An advanced data mining analysis of the testing time and the trend of test incidences is included. This work also explores the complex problem of the influence of various incidences on the aircraft testing time. The core application developed in R is supported by an easy-

[*]Corresponding author.

Email addresses: fbiscarri@us.es (Biscarri, F.), imonedero@us.es (Monedero, I.), dlarios@us.es (Larios, D.F.), jbarbancho@us.es (Barbancho, J.)

to-use customer web application using the Shiny environment. For both the data mining process and the figures shown, simulated data have been used. Currently, the framework developed is used by Airbus DS with the actual data of its aircraft. Additionally, the framework was applied in real-world cases of tests to be performed by Airbus DS, producing a useful decision-tool for company experts to evaluate the ground testing sequence.
Keywords: Information processing, Data fusion, Decision aid, Instrumentation, Test facilities, A400M Aircraft

---

1. Introduction

Airbus is a global leader in aeronautics, space and related services. In 2018, Airbus generated revenues of €40 billion and employed a workforce of approximately 55,000. Airbus Defense and Space (Airbus DS) is a division of the Airbus Group formed by combining the business activities of Cassidian, Astrium and Airbus Military. The new division is Europe's number one defense and space enterprise, the second largest space business worldwide and a top-ten global defense enterprise. Airbus DS employs approximately 40,000 workers, generating revenues of approximately €14 billion per year. The framework described in this work was developed within the project of "FSP20: Futuro Sistema de Pruebas, Visión 2020", in which our working group from the Electronic Technology Department at the University of Seville (Spain) worked for the Airbus DS Company. The FSP20 project was a collaboration among Airbus DS and various companies, as well as research organizations. The project arose and was developed in Seville, as it was the city where the process of assembly and testing of A400M was performed. The primary objective of the work described in this article was to develop a framework for the study of A400M's (Military Transport Aircraft) ground testing evolution. The A400M aircraft is currently the most important and largest military aircraft, as well as a flagship European collaborative procurement project. Thus, we can find several recent articles on Airbus aircraft's in the research literature [9, 3, 2]. However, none is focused on the topic of our work. A hundred development and certification ground tests must be performed before the A400 aircraft's first flight [8, 5]. The objective of a ground test in test facilities is the integration of a new system, and robustness testing to test fail safe functions. The greatest challenge was to be able to anticipate or modify the test scheduling to avoid impacting the final assembly. To develop the kernel of the framework, a data mining process was performed using the database and the data recorded for the test system that the Airbus DS Company uses in the ground tests of A400M. A review of the research literature shows that although aircraft's management is a fairly widespread topic of research articles [1, 6, 7], no studies specifically applicable to test management of aircraft could be found. The ground test facilities include the following:

- The Component, equipment and software test and the test of vendor parts (i.e., the test of mechanical or hydraulic components).

- The Subsystem test of software integration. The software application is linked to a hardware system for interconnection and communication with the aircraft (e.g., software integration benches are used to integrate the software in an aircraft computer; the Unit Under Test (UUT) is the aircraft computer, the power supply, cooling, and patch panels).

- The System Integration tests performed at conditions as close as possible to those of realistic flight operations with air loads and sensor inputs for the full system and a simulation of a realistic flight profile.

Why test this way in Ground Test Facilities? First,several test objectives are mandated by law or certification rules. The second reason is economic. The goal is to find as many deviations as possible from the outset during an early stage test. Early validation phases have manageable functions and clear definitions of responsibilities. It is clear that ground testing is an essential yet also expensive process performed by Airbus DS in the manufacturing cycle of an aircraft. All parts and systems of the aircraft must be meticulously checked in this type of tests; this process involves a large number of company staff and a significant amount of resources. When testing of an aircraft is performed, it is usual for incidences to occur throughout various tests. These incidences in the test process imply delays in the deliveries of the aircraft as well as additional costs for the company. This is due to tests with incidences having to be repeated (totally or partially), which generates an increase in the number of hours spent by workers and an extension of the time needed to perform the test. This paper explores the management of the sequence of tests. Additionally, it covers the influence of failures (known in the terminology of the company as "incidences") on the aircraft testing time. All tests' implied incidences, defined as test faults, are considered. One or more incidences in a test imply that the test will be repeated. As a matter of sequencing, it is possible that two tests will be run simultaneously for a certain period of time; however, a test would often await the ending of the previous test. Thus, the framework shown in this paper helps the company experts optimize the process of test sequencing and reveals the presence of a critical test from the point of view of incidences. Additionally, in a finding that is even more interesting for the company, the framework shows critical test and critical incidences in a test that suggest to a company expert the actions to be performed such that the incidence occurs neither in the current aircraft nor in the successive ones. Specifically, the article describes a framework to help a company expert establish and modify over time the sequence of ground tests and understand the influence of every test fault (incidence) on the A400M ground testing time. This framework was developed using an R software-based application that includes a clear data presentation, interactive plots and an advanced data mining analysis of the testing time and the trend of test incidences. The core R application is supported by a web application using the Shiny environment. Shiny is an R package that makes it easy to build interactive web apps using R directly. It combines the computational power of R with the interactivity of the modern web. R together with Shiny is a free software environment for statistical computing and graphics. Thus, a company expert only needs a free web browser to execute the presented framework. In this article, the proposed framework is described in the following sections:

- Data extraction and data analysis.
- The framework environment
- Results and conclusions.

Throughout the article, there are several figures in which certain data (such as the names of stations, the user codes, the test codes or the codes of the aircraft) have been redacted due to a confidentiality agreement between Airbus DS and the University of Seville.

2. Data extraction and data analysis

For the data mining process, a dump of the simulation database, as well as files extracted from the folder tree of the test system, were used as the data source. The test system includes a data integration systems [4] that provide transparent access to a collection of data stored in multiple, autonomous and heterogeneous data sources.

The data related to simulations condense all the information recorded prior to the real-world test of the airplane. The data used for the folder tree included both the recorded files with the results of various tests in the past and the source code of tests executed on those aircraft. It is not possible to provide many details for Airbus' data due to the confidentiality required by the Airbus DS Company. Preliminary data exploration was programmed with the IBM SPSS Modeler tool. This tool is a software application for data mining and text analytics developed by IBM and used to perform analytic tasks. It has a visual interface that allows users to leverage statistical and data mining algorithms without programming. The aim of the data mining process was to analyze the data source to manage:

- The test sequence (the relative location of each test in the full test sequence),
- The classification of the test (i.e., incidence, warning or success).
- The timing of the test (the start and end dates of a successful test).
- The influence of external factors onthe test (the aircraft code, the type of test, the technological area, the number of signals defined for the test, the test length or the number of sections of the test, the station and workstation, the concurrence or the number of executions of other tests in parallel, and the identifier of the user who executed the test).

Specifically, Airbus DS considers two types of incidences:

- Abortive incidences or test failures during execution. The occurrence of this type of incidences implies the failure of the test. These incidences are due to serious failures in the course of the testing process. Therefore, the prediction of this type of incidences was of the highest priority because they caused higher costs to the company than did other incidences.

- Non-abortive incidences or warnings: This type of incidence does not imply the mandatory termination of the test; therefore, the operator may continue with the execution of the test if an incidence of this type has occurred. Subsequently, the operator must correct the failure that generated the incidence and execute only the section of the test that produced the incidence.

2.1. Description of data

In this first phase, all available information was processed for each test execution extracted from the simulation environment,and the preparation of these data was performed. To this end, a set of tables from the database of Airbus DS was used as well as the information regarding the historic files for the tests. The historic files are a set of text files generated in the course of the execution of a test that record information on variables' values and relevant events as well as data on the recorded incidence (if an incidence occurred during that execution). For the shown framework, the information from the simulation data for 8 aircraft has been used. Only the fully completed tests have been included in the sample. We use the results for these aircraft considered by the Airbus DS Company to be the most representative to date. A program was developed in the IBM Modeler to process the collected data and generate a table that condensed and recorded all the useful parameters necessary for the description of the test and incidences (Table 1). Each row of the table recorded the information for a test execution performed in the past.

Table 1: Test data

Parameter	Description
TI	Code that unequivocally identifies each test
MSN	Code of the A400M aircraft for which the test was executed
ATA	Standard code of the type of test of the corresponding TI
AREA	Code representing the technological area (MTISA, MTISC, MTISN or MTISY) of the test
GTI	Code that specifically identifies the primary test of a specific plane. Every GTI includes several executions (TESTCODE values)
TESTCODE	Code that identifies the particular execution of a test on a specific plane
STARTP	Date and time of start of test execution
FINISHP	Date and time of completion of test execution
DURATION	Duration, in hours, of the execution of the test
STATION	Station of the Airbus DS factory in which the execution of the test was performed
WORKSTATION	Computer of Airbus DS on which the test system was executed to perform the test
DAYS_STATION	Days that have elapsed since the first test of that aircraft was executed on the station
CONCURRENCE	Number of executions of other tests that were executed in parallel with the test
OPERATORS	Number of operators necessary for the execution of the test
USER	Code of the primary user that performed the execution of the test
INC_A	Flag that recordswhetherthere was an abortive incidence (value of 1) or not (value of 0) during the test
INC_W	Flag that registers if the test resulted in any non-abortive incidences or not
INC	Flag that records whether the test involved an incidence of any type (value of 1) or not (value of 0). Thus, this flag identifies if the test involved an abortive incidence or a non-abortive one
INCIDENCE_TYPE	The type of abortive incidence (if one was recordedduring the corresponding test)
DESCRIPTION	The description of the abortive incidence (if one was recorded during the corresponding test)

2.2. Data analysis

In this second phase, an analysis of the data was performed.The initial distribution of the total sample with regard to the total number of test executions performed to date for the 6 aircraft considered in this study as well as the number of executions with incidences of each type were analyzed. Thus, the sample set had a total of 16, 095 test executions.

3. Framework environment

As a software decision support tool, the shown framework has been developed as a Microsoft Windows application. Access to the program's functions is divided into independent modules (tabs). The primary tab (MSN Explore) presents the following user interface (Figure 1).

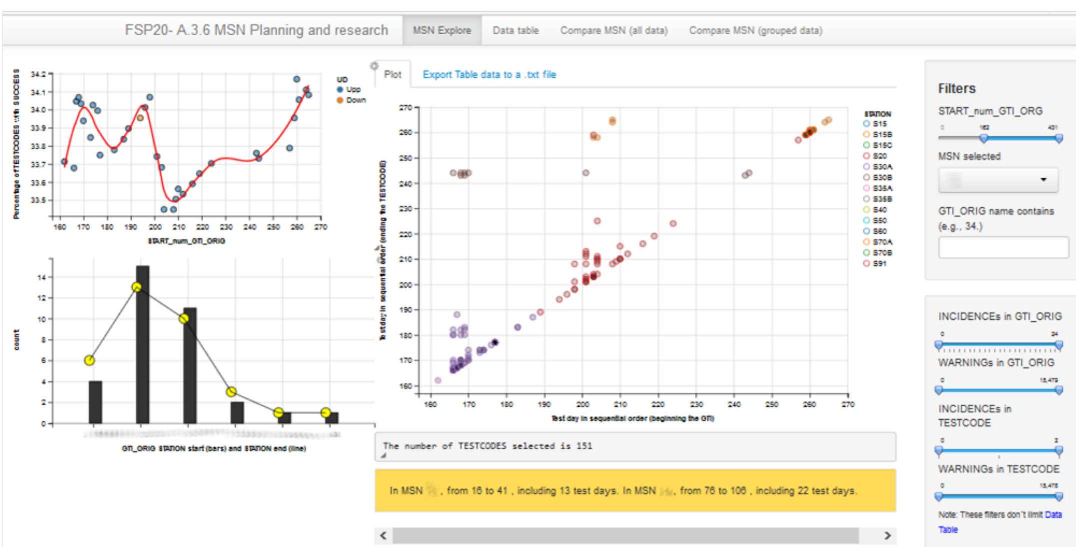

Figure 1: First tab: MSN Explore (simulated data)

The framework is an application developed using the software package called Shiny, a web application supported by R. It is not necessary to connect the computer to the Internet to run this software. This aspect is important due to the need to maintain the privacy of data. The program execution opens locally the default web browser. Shiny is an R package that builds interactive web apps directly using R. All plots or tables are interactive and easily exportable by clicking on the object ”controls” with the mouse (Figure 2).

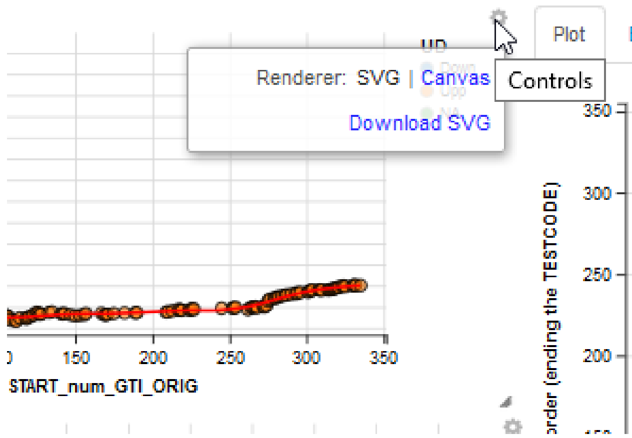

Figure 2: All framework’s elements are exportable objects

The program is a client-server runtime process. A customer may execute simultaneously any desired number of client instances.

3.1. First tab (MSN Explore). Primary plot (right side)

The primary plot shows ”Test days in sequential order (the ending TESTCODE)” plotted versus ”Test days in sequential order (the beginning GTI)”. The x-axis, showing the day that the GTI began, represents the time sequence of the primary group of tests. The y-axis, showing the day that the every execution ended, represents all the executions for every GTI and the duration of this set of executions (Figure 3).

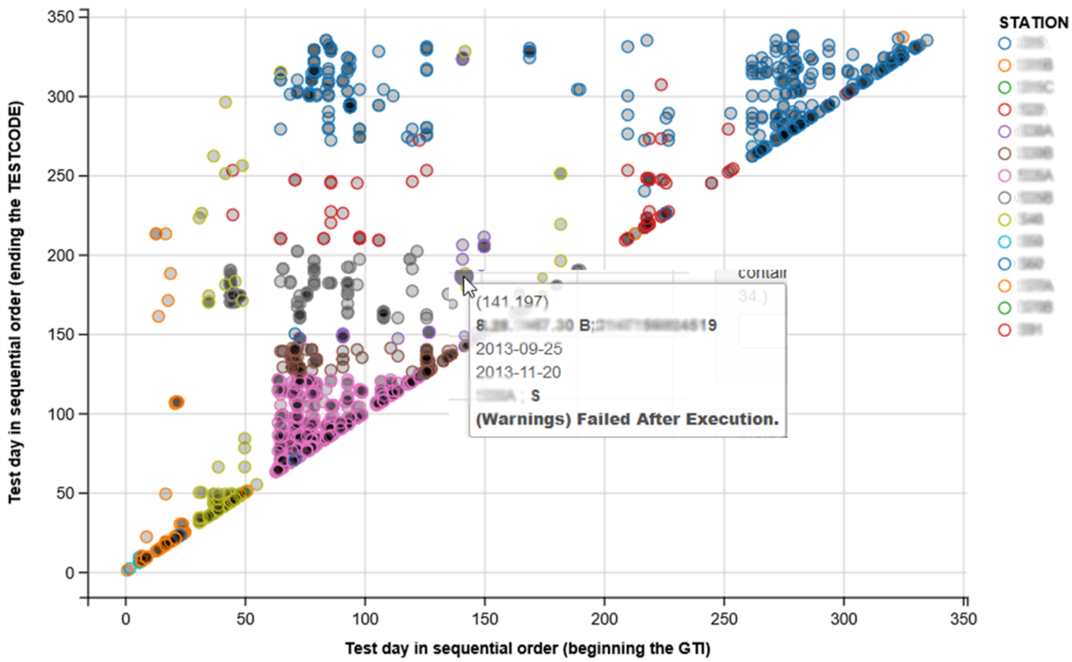

Figure 3: tab: The primary plot (simulated data)

When the mouse cursor hovers over a plot's point, the interactive plot shows that the GTI number XXXXX.30 B began on 2013-09-25 (day 141 of the aircraft's test set sequence). The execution shown (TESTCODE) has the XXXXX9 code and ended on 2013-11-20 (day 197 of the aircraft's test set sequence) on a specific test station, with the result of non-abortive incidences or warnings that did not imply the mandatory termination of the test. The final conclusion is the success of this execution. Below the primary plot, the total number of executions selected for this aircraft is shown (Figure 4).

```
The number of TESTCODES selected is 1894
```

Figure 4: Automated text notice (simulated data)

The option "Export table data to a.txt file" (included in this plot) allows Airbus' DS expert to export (to Microsoft Excel, by clicking on "export data to.txt") and visualize all data contained in historic files of all the test's executions (sections of test, types of warnings and incidences, etc.). This option allows experts to explore the test set in detail. The name of the exported file includes the date it was exported.

3.2. First tab (MSN Explore). Data filters

On the right side of the first tab of the framework, a set of filters allows an Airbus DS expert to define a selection of data, limiting the number of incidences and/or warnings in GTIs, and the number of incidences and/or warnings in TEST-CODEs. Additionally, the aircraft being tested (MSN) is selected, and the possibility of selecting only certain GTIs by name is available (Figure 5).

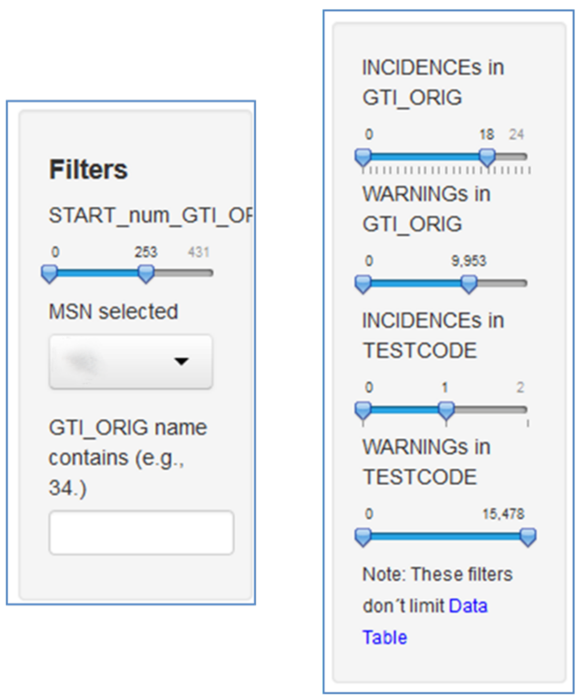

Figure 5: Global data filters in the first tab

3.3. First tab (MSN Explore). Plot of the evolution of success in test executions

The cumulative percentage of successful executions is a basic parameter used to measure the quality of aircraft testing. The evolution of testing success is shown (Figure 6).

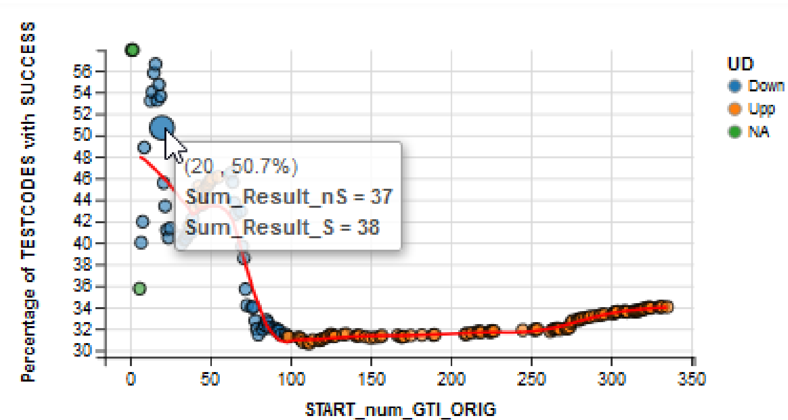

Figure 6: First tab. Plot of the evolution of the number of successful executions (simulated data)

The x-axis shows the day the GTI began (the first execution day of the TEST-CODE of the GTI). This axis represents sequential days of an aircraft test (e.g., the first day of test, the second day). The y-axis shows the cumulative number of executions completed successfully. Thus, the point selected interactively indicates that from day 1 to day 20 of aircraft testing, 38 TESTCODEs with SUCCESS and 37 TESTCODEs without SUCCESS have been recorded. This result implies that 50.7% of executions were successful from day 1 to day 20. The temporal evolution of this percentage has been measured by means of the slope of the average of data. This Up Down parameter indicates the slope showing an increase (the section of sequential days with the percentage increasing) or a decline (the section of sequential days with the percentage declining). Blue points on the plot indicate the downdays, and orange points indicate updays. An automated message obtains the algorithm's result and informs the company expert of the results of downdays (Figure 7). It is an automatic warning message.

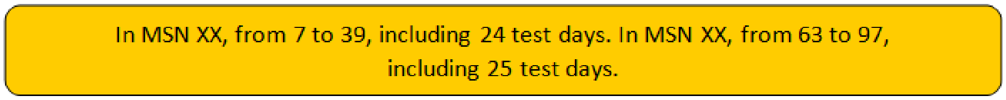

Figure 7: First tab. Warning message (simulated data)

3.4. First tab (MSN Explore). Plot of the evolution of executions in Test Facilities' Stations

Another interesting parameter according to the experience of Airbus DS' experts is the evolution of tests at different facilities' stations. Usually, an execution begins and ends at the same station; however, various problems necessitate a station change. The knowledge of these changes helps experts detect stations with an excess of tests (compared to others) or establish a station classification by the percentage of station's test success (Figure 8).

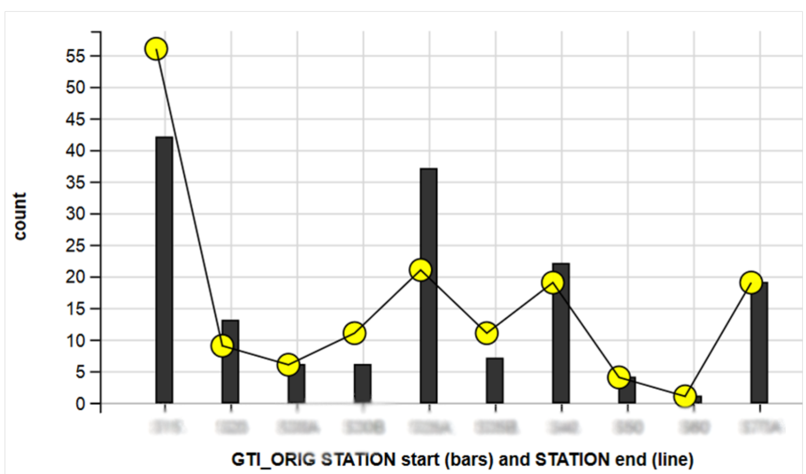

Figure 8: First tab. Evolution of executions in Test Facilities' Stations (simulated data)

3.5. Second tab. Section Data Table

The second tab allows an Airbus DS expert to visualize all data collected in the database of historical executions of aircraft testing. Columns to be shown may be easily selected. The framework contains a fast filter that selects a text string that any cells in the table must include. A long text string implies more specific information (Figure 9).

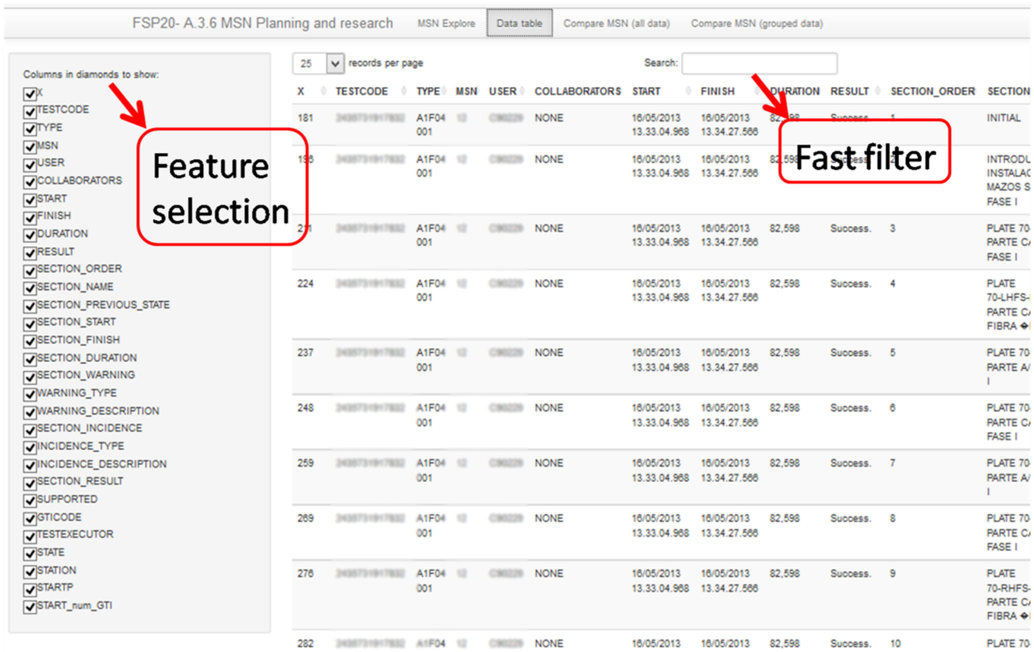

Figure 9: Second tab. Interactive Data table (simulated data)

3.6. Third tab. Section: Compare MSN (all data)

This screen shown four aircraft's data for simultaneous comparison. Features to be plotted along the x-axis or y-axis are selected using the combo box on the right side. Airbus DS experts have suggested the following primary axis features: TESTCODE date, Station, Incidences in GTI, Incidences in TESTCODE, Warnings in GTI and Warning in TESTCODE. A company expert could compare not only dates, warnings and incidences in GTI or in TESTCODEs but also the evolution of testing of different aircraft or the station testing performance. Two additional filters have been included to select the result of execution (with or without success) and the date range (Figure 10).

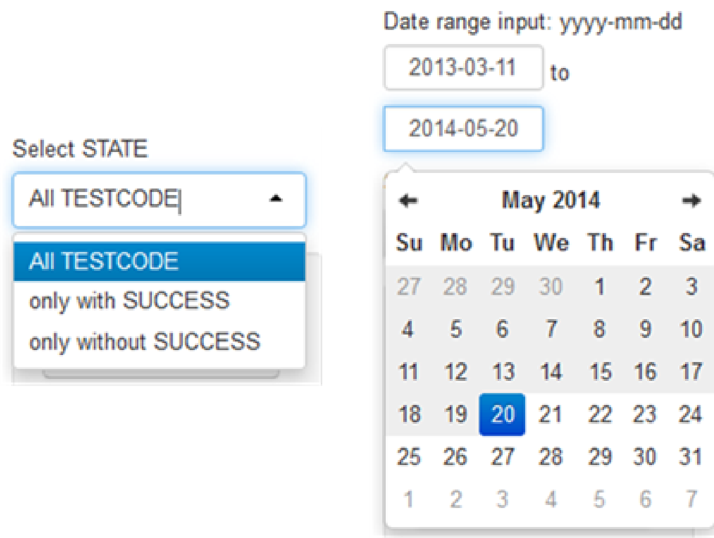

Figure 10: Third tab. Additional filters

By way of an example of the possibilities of this screen, an expert can make the following selection: the X-axis and Y-axis both show the "TESTCODE date"; the filter of the Selected STATE shows records "only with SUCCESS" (Figure 11).

The Airbus DS staff visualize and compare the days on which successful tests have been executed. The real-world successful test sequence (not the proposed test sequence) is shown. Thus, for instance, the GTI coded 8.XXXXXF has been executed the first time on 2013-09-24 (day 198 of the real-world test sequence) and was finished successfully on day 305 of the test (2014-01-09,with the TESTCODE of 4XXXXXX7). The real-world sequence of successful tests may be easily checked.

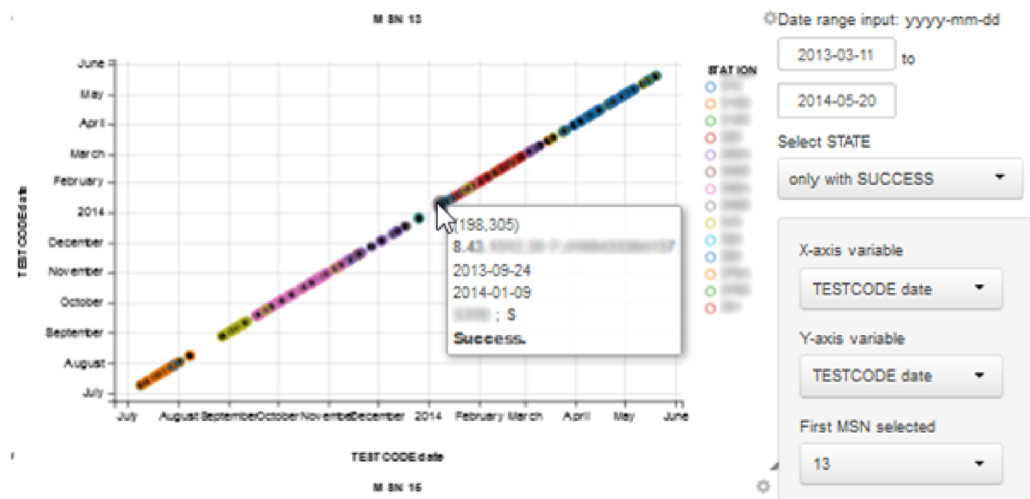

Figure 11: Sample selections and the resulting chart. Sequence of successful testing (simulated data)

3.7. Fourth tab. Section: Compare MSN (grouped data)

The third tab allows the comparison of all GTIs and TESTCODEs among four different aircraft. The fourth tab allows the comparison between different aircraft of the incidences and warnings accumulated in one day, one station or one chapter according to ATA. Thus, for example (Figure 12), the framework detects the maximum number of incidences (15) during a testing day of an aircraft with a specific MSN (day 213 of the testing sequence, 2013-10-09). This occurred on the GTI identified by 8.XXXXF that was successfully completed during the execution (TESTCODE) 23XXXX54 in the XXA station.

4. A data minig framework for aircraft's test management

Thus, the presented framework (Figure 13) starts from data integration and it covers the following items:

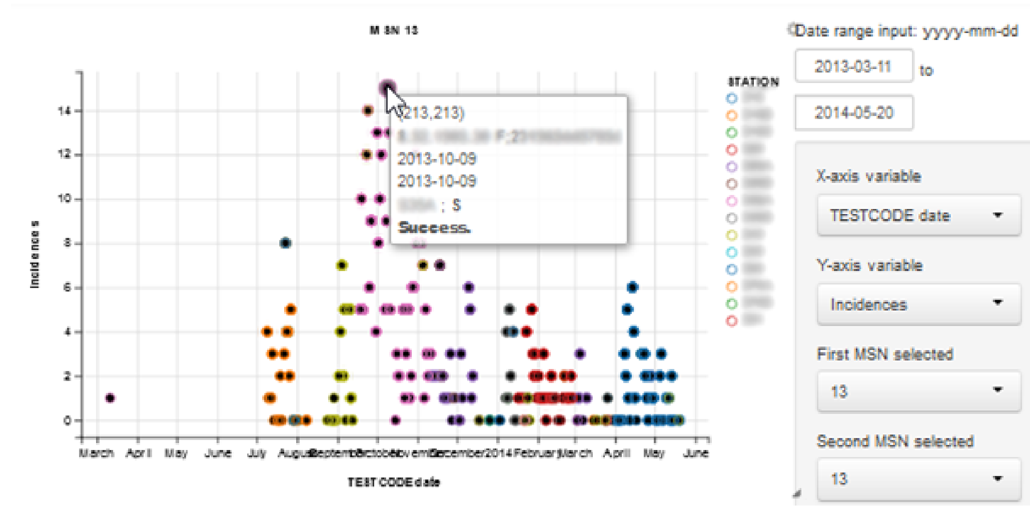

Figure 12: An example. Maximum number of incidences during a testing day (simulated data)

- The exploration of the test secuence. Searching for critical test's days, this process allows experts to manage the sequence of the test, to change some test days or to change the test's station. Last day of any test contains relevant information (Figure 3), but the slope of the smoothed sequence also reveals critical problems, grouping days with more o less successful executions. This measure has been performed thanks to Loess Regression method, that smooths a volatile time series. Loess (short for Local Regression) is a non-parametric approach that fits multiple regressions in local neighborhood. This can be particularly resourceful, if you know that your X variables are bound within a range. R software use the loess() function on a numerical vector to smooth it and to predict the Y locally (i.e, within the trained values of Xs). The size of the neighborhood can be controlled using the span argument, which ranges between 0 to 1. It controls the degree of smoothing. So, the greater the value of span, more smooth is the fitted curve. For finding the optimal value of span,

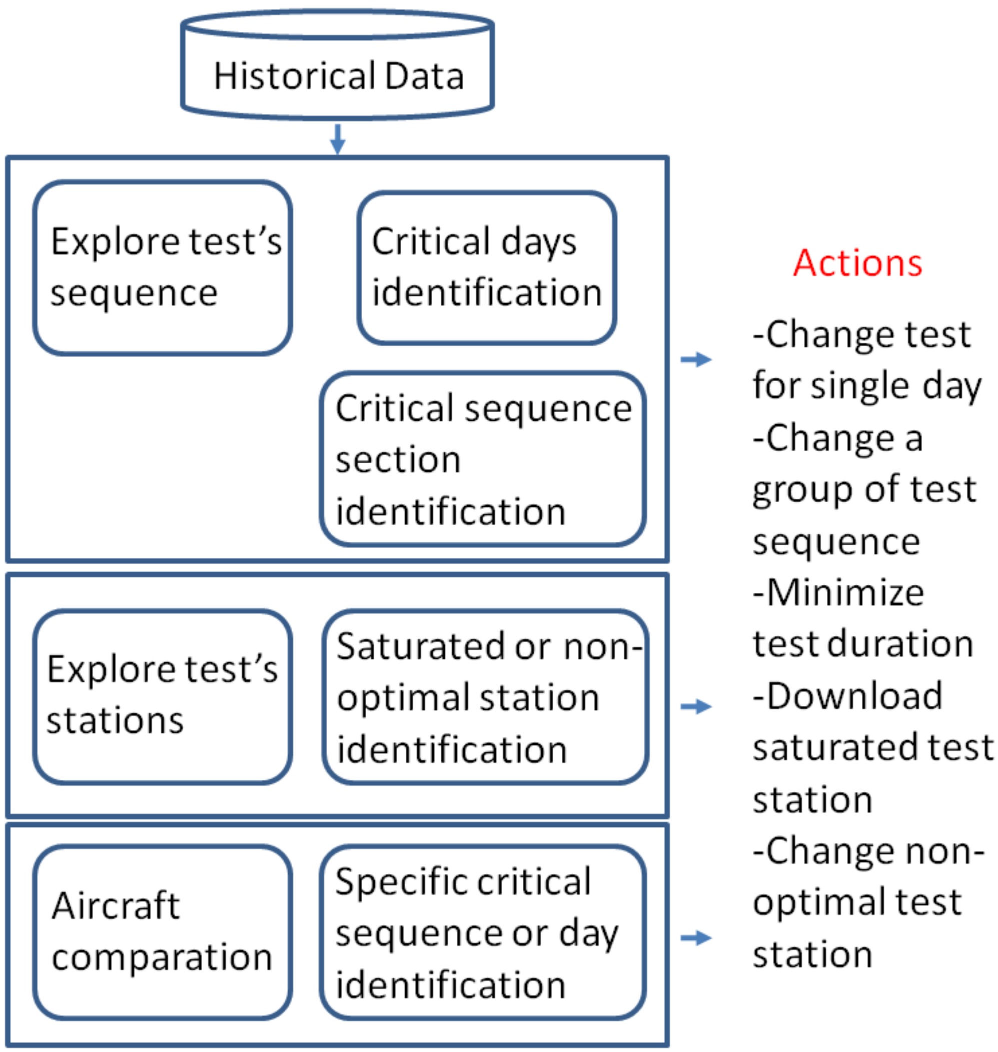

Figure 13: A data minig framework for aircraft's test management

the Sum of Squared Errors (SSE) are minimized. Thus, in this research, span is fixed to 0.3 (Figure 6).

- The exploration of test's executions in stations. The exectution of several tests in different stations and the reduction of the number of test on a saturated station improve the test's sequence execution (Figure 8).

- The comparison between different aircrafts. Using all data, the historical test data improve future test executions on new aircrafts (Figure 11). With grouped data,a specific problems attached an specific aircraft would be detected (Figure 12).

5. Results and conclusions

The incidences in the test process of an aircraft imply delays and costs for the company. Our team from the Electronic Technology Department (Spain) worked with the Airbus DS Company and developed a framework for incidence prediction using a data mining process. Specifically, the framework is designed as a decision support expert software for the ground testing sequence management of the A400M aircraft. It was not possible to find any work in the research literature that examined specifically the topic of using data mining to improve the testing sequence in the context of aeronautics, where this article makes an important contribution. The primary features of the proposed framework are:

- Providing an easy, secure, interactive and R-powered management of the full test set of a group of aircraft. Thousands of tests are readily accessible.

- The clear knowledge of the temporal order of the successful test when facing an ad hoc proposed testing sequence.

- The exploration of the length of every test, the test length comparison and the influence of incidences and warnings of each test of the full test sequence. Furthermore, the presented framework highlights the importance of changes in the test sequence.
- The fast visual detection of tests with a high number of incidences or warnings and their causes.
- The comparison of a large number of test features among several aircraft's testing results using all data or grouped data.
- Defining new parameters to measure the accuracy of a test sequence as the cumulative percentage of successful executions.

Although the framework has been designed and tested with simulated data, it is currently in the last stage of testing by the Airbus DS ground test staff using real-world data, while the next step involves its installation on the computers of test planners of the company.

Acknowledgment

This work was performed as part of the project "FSP20: Futuro Sistema de Pruebas, Visión 2020" funded by the Center for Industrial Technological Development (CDTI) under the FEDER-INNTERCONECTA program. The authors thank the Airbus DS staff for their collaboration as well as for providing the facilities at the time of publishing this work. Specifically, we would like to thank AG and JM (removed due the double-blind review policy) for their invaluable work and collaboration throughout the research process.

References

[1] Appriou, A., 2007. Special issue sensor management in complex systems. Aerospace Science and Technology 11, 259 – 260. doi:https://doi.org/10.1016/j.ast.2007.02.004. cOGIS '06.

[2] Gleyzes, C., Capbern, P., 2003. Experimental study of two airbus/onera airfoils in near stall conditions. part i: Boundary layers. Aerospace Science and Technology 7, 439 – 449. doi:https://doi.org/10.1016/S1270-9638(03)00045-2.

[3] Hernández, S., Menga, E., Naveira, P., Freire, D., López, C., Montoya, M.C., Moledo, S., Baldomir, A., 2018. Dynamic analysis of assembled aircraft structures considering interfaces with nonlinear behavior. Aerospace Science and Technology 77, 265 – 272. doi:https://doi.org/10.1016/j.ast.2018.03.004.

[4] Maw, A.A., Lwin, T., Tyan, M., Lee, J.W., Kim, S., 2018. Efficient approach to database integration for an aerospace vehicle design and certification framework. Advances in Engineering Software 118, 27 – 34. doi:https://doi.org/10.1016/j.advengsoft.2018.01.001.

[5] Nicolosi, F., Marco, A.D., Sabetta, V., Vecchia, P.D., 2018. Roll performance assessment of a light aircraft: Flight simulations and flight tests. Aerospace Science and Technology 76, 471 – 483. doi:https://doi.org/10.1016/j.ast.2018.01.041.

[6] Rehage, D., Carl, U.B., Vahl, A., 2005. Redundancy management of fault tolerant aircraft system architectures – reliability synthesis and analysis of degraded system states. Aerospace Science and Technology 9, 337 – 347. doi:https://doi.org/10.1016/j.ast.2005.02.002.

[7] Schäfer, L., García, S., Mitschke, A., Srithammavanh, V., 2018. Redundancy system design for an aircraft door management system. Computers and Operations Research 94, 11 – 22. doi:https://doi.org/10.1016/j.cor.2018.02.005.

[8] Shang, Y., Liu, X., Jiao, Z., Wu, S., 2018. An integrated load sensing valve-controlled actuator based on power-by-wire for aircraft structural test. Aerospace Science and Technology 77, 117 – 128. doi:https://doi.org/10.1016/j.ast.2018.02.030.

[9] Viúdez-Moreiras, D., Martín, M., Abarca, R., Andrés, E., Ponsín, J., Monge, F., 2017. Surrogate modeling for the main landing gear doors of an airbus passenger aircraft. Aerospace Science and Technology 68, 135 – 148. doi:https://doi.org/10.1016/j.ast.2017.04.021.