Knowledge Management for Propulsion Systems Integration

by Matthieu GONSOLIN

gonsolin@kth.se

October 2012 – March 2013
Abstract

This report details my work as the endpoint of the master's program in aerospace engineering that I attended at the Aeronautical and Vehicle Engineering Department of Kungliga Tekniska Högskolan, Sweden. It is as well the conclusion of my internship in the Propulsion Systems Integration domain (EPT3) of Airbus Operation SAS, France.

On the one hand, airlines order new planes and the worldwide fleet increases, while, on the other hand, the market pressure, the rise of fuel prices and other factors contribute to regular changes in the technology. These drivers may impact maintenance activities and support to operators, and the number of issues occurring on in-service aircraft. In-service and production queries are a specific type of support activities followed-up by propulsion systems integration engineers from the aircraft manufacturer, such as Airbus. These technical questions can address any of the engine's systems and must usually be answered to within a short timeframe as they might delay a flight or the delivery of an airplane. In the global scope of knowledge management inside the company, these engineers realized their loss of not capitalizing these activities and promoted this project. An adapted application has been developed to share the experience among programs and support the engineers for the treatment of such queries. As the focus of the project was put on assessing the actual need of the future users to provide an adapted tool, the database should prove its performance over the long term. This paper details the different steps of the project: analysis of the need, specifications, programming and testing, that led to meeting this specific need for capitalization.
Acknowledgements

At the end of this thesis, I would like to take the time to express my thanks to all those who contributed in one way or another to the accomplishment of this project and made it a very valuable experience, as much for my personal development as by increasing my working skills and knowledge.

Foremost, I would like to express my sincere gratitude to my supervisors.

I truly thank Mr. Alexandre Armand, my company tutor for taking the responsibility of an intern in addition to his own work and for taking the time to guide me throughout the project, to teach me plenty about propulsion integration and Airbus’s structure and activities. His interest and involvement in the project are one of the main drivers that led to the commissioning of the functional application used today by all the teams.

I truly thank Prof. Björn Laumert, my university tutor for accepting to supervise my thesis and give me valuable feedback and evaluation of my work throughout the project.

I am thankful to Mrs. Sylvie Michel and Mr. Eric Desmet, the section and sub-domain heads for being supportive to the project and being attentive to my propositions. A hierarchy back-up is always essential to the commissioning of a new tool and they greatly contributed to establishing its specifications. I hope that my work will meet all their expectations in the long run.

I take the opportunity to sincerely acknowledge Kungliga Tekniska Högskolan for providing the academic support essential to making this thesis project possible; and the Airbus Company for providing such an internship opportunity. I am also thankful to Airbus for their material support and their dedication to create a favourable work environment for engineers.

I am grateful to the other sections’ managers, Mr. Olivier Pelagatti and Mr. Laurent Mistral, for their involvement in the project and their useful advice and feedback. I hope that my application will be as great a help to them as they were for me during its development.

I want to give special thanks to the EPT31 team. My former colleagues integrated me right away and their warmth and cohesion made me feel welcome, which contributed in a way to improve my work.

Finally, I would like to acknowledge every other person in Airbus that contributed to develop this application; by giving their opinion and comments, by showing their interest and by sharing their knowledge and experience with me. I will treasure these six months within Airbus as a truly fulfilling experience.
# Table of contents

Abstract ........................................................................................................................................... 2
Acknowledgements .......................................................................................................................... 3
Nomenclature .................................................................................................................................... 6
List of figures and tables .................................................................................................................... 7
Introduction ....................................................................................................................................... 8

I. Objectives ................................................................................................................................... 9
   A. Introduction to knowledge management .................................................................................. 9
      1. A difficult concept .............................................................................................................. 9
      2. State-of-the-art .................................................................................................................. 10
   B. Motives of the project .......................................................................................................... 10

II. Method of attack and methodology ............................................................................................ 12
   A. Task division and timeline .................................................................................................... 12
   B. Establish a new work process .............................................................................................. 13
   C. Establish the specifications .................................................................................................. 14
   D. The different steps of programming ..................................................................................... 15
   E. Commissioning ..................................................................................................................... 17

III. Theory and background ............................................................................................................ 18
   A. Technical information ........................................................................................................... 18
      1. The aircraft engine: a sophisticated system ..................................................................... 18
      2. Technical issues ................................................................................................................ 27
   B. Work processes .................................................................................................................... 27
      1. Continuous airworthiness and development .................................................................... 28
      2. Continuous support ........................................................................................................ 29
      3. Customer service ........................................................................................................... 30
   C. Company organizational structure ....................................................................................... 30
      1. Coordination between the airlines and manufacturers ..................................................... 31
      2. Center of Competence powerplant .................................................................................. 31
   D. Customers and suppliers: an intertwined structure ............................................................... 33
   E. Knowledge management in EP ............................................................................................. 36
      1. Lack of adapted tools ........................................................................................................ 36
      2. A need for shared knowledge ........................................................................................... 36
IV. Implementation

A. Specifications in accordance with the users’ expectations
1. High level requirements
2. DBMS and spreadsheets
3. Low level requirements

B. Programming under Excel
1. Object-oriented programming
2. Excel class diagram

C. Developing a user-friendly application
1. Presentation of Tanga_DB
2. Programming choices and highlights of the code

D. Filling the database
1. Final debugging
2. Trace past experience

V. Results

A. Simulation of the new process
1. Receipt of the query
2. Searches through Tanga_DB
3. Fill the database

B. Performance

Conclusion

References

Appendices

* Confidential
Nomenclature

AMM Aircraft Maintenance Manual
AOG Aircraft On Ground
API Application Programming Interface
APU Auxiliary Power Unit
ARQ Airline Routine Question
ATA Air Transport Association
CMM Component Maintenance Manual
CoC Center of Competence
COM Component Object Model
CU Cockpit Unit
DBMS DataBase Management System
DFDR Digital Flight Data Recorder
DMU Data Management Unit
EASA European Aviation Safety Agency
ECAM Electronic Centralized Aircraft Monitor
ECU Engine Control Unit
EEC Electronic Engine Controller
EGT Exhaust Gas Temperature
EIS Entry Into Service
EVMU Engine Vibration Monitoring Unit
FADEC Full Authority Digital Engine Control
FAL Final Assembly Line
HMU Hydro-Mechanical Unit
HPC High Pressure Compressor
HPT High Pressure Turbine
HPTCC High Pressure Turbine Clearance Control
IAE International Aero Engine
IDG Integrated Drive Generator
ISRO In-Service Reportable Occurrence
IKM Innovative Knowledge Management
KM Knowledge Management
LPC Low Pressure Compressor
LPT Low Pressure Turbine
MAP Mise Au Point
MISP Major In-Service Problem
MOD MODification
MSN Manufacturing Serial Number
N2 LPT shaft’s speed
OGV Outlet Guide Vane
OOP Object-Oriented Programming
PATM Production Aircraft Test Manual
PERT Program Evaluation and Review Technique
PPS Power Plant System
PSI Propulsion Systems Integration
RA Risk Analysis
RFW Request For Work
RISE Reuse, Improve and Share Experience
RTD Resistive Thermal Devices
SA Single-Aisle
SAV Starter Air Valve
TA Technical Adaptation
Tanga_DB The application
VBA Visual Basic for Application
VBV Variable Bleed Valve
List of figures and tables

Figure I.1, KM solution portfolio

Figure II.1, PERT chart
Figure II.2, Gantt chart
Figure II.3, New work process
Figure II.4, V-cycle for software programming

Figure III.1, Engine principle and flow path
Figure III.2, Fan components
Figure III.3, CFM56-5B with systems
Figure III.4, Starting and ignition system components
Figure III.5, Variable bleed valves
Figure III.6, HPTCC valves
Figure III.7, Lubrication system components
Figure III.8, Oil tank components
Figure III.9, Fuel distribution system components
Figure III.10, Nacelle components
Figure III.11, Engine Mounts
Figure III.12, Starter transmission housing
Figure III.13, Thrust face wear on cluster gear
Figure III.14, Safety keep-out zone
Figure III.15, MSN 5312, IAE Engine vibration in during second flight
Figure III.16, Interaction engine / aircraft
Figure III.17, Integration activities
Figure III.18, Organization structure of EP
Figure III.19, Organization structure of EPT
Figure III.20, A role of coordination
Figure III.21, Joint ventures between engine manufacturers
Figure III.22, Main nacelle manufacturers
Figure III.23, Motorization of the A320 Family

Figure IV.1, Excel class diagram
Figure IV.2, Properties of an Item in Tanga_DB
Figure IV.3, Structure of the tool: Worksheets and Userforms
Figure IV.4, Tanga_DB - Home page
Figure IV.5, Tanga_DB - Database
Figure IV.6, New_Item form
Figure IV.7, Management of the mandatory fields
Figure IV.8, Search_Item form
Figure IV.9, The “Sort” function
Figure IV.10, Details_Item form
Figure IV.11, Ink Edit control

Figure V.1, Results of the search
Figure V.2, New query in Tanga_DB
Introduction

One can remember the massive crisis that struck the world economy back in 2008, leaving a stunted growth and a devastated economy which took years to rebound. And the Education financial bubble in the United States is now threatening to burst in a near future and expected to have even greater a fallout; while many countries in Europe are already struggling towards economic recovery and the reduction of their debts through one more austerity plan. However, despite all the downturns experienced in the last decades, commercial aviation has shown a notable resilience and still foster growth. A survey commissioned by Airbus recently reveals that among “10,000 people questioned worldwide, the majority believes that we would fly more in the future” [1]. Boeing states on its forecast release that the world passenger traffic should grow by 5 percent annually over at least the next 20 years. These market trends can already be appreciated at the aircraft manufacturers’ level since most of the order books are full. As of September 2012, Airbus has yet to deliver more than 4,000 aircraft (all families considered) to committed customers. The involvement of the largest companies in new aircraft programs, such as the A350-XWB, the A320neo or the Boeing 777X, along with the tightening presence of the Brazilians from Embraer, the Canadians from Bombardier, the Russians from Sukhoi and the Chinese manufacturers on the market, is clearly another factor demonstrating the dynamism of the commercial aviation industry. The overall fleet cruising through the sky is then enlarging and so are the numbers of airlines, of different planes and of engine versions. Moreover, the market pressures, the continuous rise of fuel prices, the properties of newly discovered materials and so on, induce regular changes on the technology. All these factors and the market drivers tend to amplify the amount of activity regarding maintenance and support services to operators. [2]

However, no matter which change it has undergone, no matter which configuration it is in, a plane is still a plane and an engine remains an engine. Many problems encountered by the specialists can therefore be recurrent over time, or at least similar from one to the other, hence the advantages of a time-saving resource that a well structured knowledge management tool would be are fully understandable. On the other hand, knowledge management is also nowadays a critical factor in a company’s success. This paper details the various steps towards the commissioning of an adapted computerized tool, named Tanga_DB, that will effectively enable Airbus PSI engineers to save time and resources during support to in-service (already flying) and production aircraft by capitalizing their activities. In order to be able to design such an appropriate tool, one has to think as a virtual user-to-be and fully understand the material behind the future database. We will therefore start by detailing the complex technology behind an airbreathing jet engine for commercial aircraft and the associated activities of engine integration within Airbus. Armed with this knowledge, we will be able to examine the various reasoning and choices leading to the formulation of the requirements, the conceptual design and the programming phases of the tool. The usefulness of such a tool being mainly relevant over time, we will however bypass our lack of extended use and try to offer a primary efficiency assessment of Tanga_DB in terms of actual improvements to the work load and user perception.
I. Objectives

A. Introduction to knowledge management

1. A difficult concept

It is often delicate to define properly the concept of knowledge management since there exists no universal definition of it and it refers to “knowledge”, which itself can almost represent anything. A commonly shared conviction is however that the success of today’s corporations is more and more conditioned by their ability to deploy optimized processes, develop better tactical problem solving and technical know-how in shorter timeframes. More than capital or natural resources, the one sure lasting asset is a well-managed knowledge, which tends to make a company stand out from the competition. Even more in the world of technology, domination ensues directly from creating, organizing, storing, sharing, selling and protecting knowledge between teams, people or organization. Michel Grundstein gave in 1995 a possible definition of knowledge management in a company: “Locate and make visible the enterprise knowledge, be able to keep it, access it and actualize it, know how to diffuse it and better use it, put it in synergy and valorise it”. This definition comprises many words, highlighting the multiple facets of knowledge management. [3]

Knowledge can take many different forms that require completely different types of management, for example:
- Product description
- Process description
- Contacts
- People
- Organization
- Web pages
- Best practices
- Methods and tools
- Handbook
- Technical documents

Some direct outcomes can be given to realize better the challenges that it represents for the company to manage this knowledge. [4] The first idea is obviously to keep a trace of all the work that has been done, in order to avoid performing twice the same tasks or errors and continuously improve the processes (Best practices, Methods and tools, Handbook, etc). The second idea ensues from retirement and an increased circulation of jobs (voluntary turnover or redundancy) that tend to leave companies with employees more often lacking job-specific knowledge that need to be compensated for by new computerized tools so that the company can sustain its competitiveness (People, Best practices, etc). [5] With the same capabilities, people with the stronger knowledge back-up will nearly always end up more productive. And yet, even if knowledge management is a key to continuous innovation, appropriate decision-making and market competitiveness; a study conducted in 2008 by the Institute for Corporate Productivity [6] showed that many companies confess still retaining knowledge poorly.

This concept of capitalization is nonetheless well deep-rooted into Airbus practices and has to be; considering that the average lifetime of an aircraft is superior to 30 years and that the company must provide continuous support and improvements to their customers. A whole department (KM) in Airbus is dedicated to knowledge management and provides a wide portfolio of tools and practices known in Europe as one of the most sophisticated programs. [7] Their motive is presented as: “Knowledge
management is based on the recognition that an organization’s most valuable resource is the knowledge of its people”.

2. State-of-the-art

Confidential

B. Motives of the project

As the life-cycle of an airplane is longer than three decades, many issues occurring on in-service engines or in production may be recurrent over time, or similar to one another. Different companies can face the same problem several months or years apart. Moreover, even if each engine features different configurations and each aircraft features different engines, the problems can be akin among programs. A new configuration can also bring back issues similar to ones treated long ago.

Production or in-service daily queries are occasional requests from airline companies or from the production lines. These technical questions can be of any type and must usually be answered to within a short timeframe as they might delay a flight or the delivery of an airplane.

A score of engineers within the domain realized their loss of not capitalizing these activities, especially since it could improve the critical factor of time response to a request. As it came within the scope of a global willingness from the hierarchy to standardize practices and share the experience among domains, the project is accepted. The idea is to develop an adapted database listing all answered daily queries to keep this information and easily access it, would a similar issue occur. The main features of this tool would be: user-friendly, simple and adapted to the needs so that the potential users would not only fill the database but also use it as a source of information.

The multiple objectives of this tool can be detailed slightly more:

- **Improve time response, accuracy and exhaustiveness of daily queries treatment**
  Enable the building of a culture of best practices based on expert knowledge from years of experience and not the engineers’ own experience only. Capitalizing the queries will allow knowing how problems have been treated in the past, which points could be improved, propose an outstanding time response if the issue has been solved previously or at least try to reduce it for completely new questions.

- **Tracing the experience for Airbus**
  As was seen earlier, knowledge is a fundamental resource for the company. This tool will allow to keep track of a knowledge that is today lost to Airbus over the long term.

- **Increase the autonomy**
  Increase the autonomy of the PSI engineer regarding the motorist but also regarding its colleagues by providing him with another source of information to spot faster where he might find relevant help, whichever form it might take (files, persons, reference, database or paper books).

- **Support to new comers**
  Reduce time transitioning employees into new roles of propulsion integration specialists and balance their lack of personal experience by providing an access to the shared experience stored in the database.

- **Less stressful environment**
  With an adapted computerized tool back-up, the engineer is more likely to solve accurately a request before the pressing deadline.
This paper details the different steps, sketching and programming, leading to the commissioning of an adapted user-friendly database enabling PSI engineers to improve their performance in the treatment of production and in-service daily queries and trace the experience for Airbus.
II. Method of attack and methodology

A. Task division and timeline

Once clear objectives have been established, another preliminary step is to draw an estimated work plan: divide the project into phases and tasks, consider the time frame and set up deadlines for the deliverables. Without needed to be fully detailed to the single task, the more the project is sequenced in a series of small steps, the more it will bring visibility and dynamism throughout it.

Figures II.1 and II.2 show two different charts commonly used in project management. The PERT chart (Program Evaluation and Review Technique) lists the main tasks and their linking. The milestones of the project are highlighted with a darker colour. The Gantt chart illustrates the project schedule accordingly. [9] Three phases were considered along with three deliverables: the specifications, the user-guide along with a prototype and the final tool.
The different phases highlighted above and the methodology envisaged in order to respect the objectives and timeline are described in the following sections with a justification of the task division in II.D.

**B. Establish a new work process**

Even before thinking about programming or defining any precise functionalities for the tool, the first and major phase of the project is to introduce the idea to the future users and most of all, to make people get involved and support the project. This way, they will contribute more actively to develop the tool with their comments and opinions so that it fits more closely their real needs. This step is the decisive phase of the project. No matter how perfect your final program may be, a computerized tool will soon turn obsolete if the users do not feel the need to use it or find it non-adapted, as they were not involved enough upstream of the commissioning. They might see the database more as a chore, another task on their full agenda than a useful source of information, established to save their time and not waste it. Working on getting the project approved by the management and adopted by the engineers is critical to the risk of not fulfilling the initial objectives or see the project dropped even before the commissioning of a prototype. The final objective is to establish a new work process adopted and followed by the PSI engineers when facing a new incoming query.
Another objective is to assess the existing methods used until now to capitalize the relevant activities in each program and the other computerized tools used by PSI engineers at Airbus (T11, T12). We will therefore learn from past experience and know what aspects to avoid and which functions to favour for Tanga_DB. This task has two different sides:

- **Management process**: How and where to get this information?
  - Interviews of PSI engineers
  - Personal hands-on sessions on the existing computerized tool
  - Exchanges with the EPDX/KM domain regarding knowledge management

- **Information**: What shall be investigated for each method?
  - Frequency of use
  - Activities covered
  - Main advantages and drawbacks (display, functions, availability, etc)

The in-groups actual methods and other computerized tools will be presented in III.E with their limitations. EPDX brought all its experience in knowledge management to the project as Tanga_DB comes within the global scope of KM in Airbus (See I.1). The domain will be presented in III.C.2.

## C. Establish the specifications

Once the in-group methods have been determined, the next phase is to start the conception of the tool. However, the objectives and the current state of the work are not sufficient to start establishing precise specifications.

A very important part of the project is to learn as much as possible about the users and their activities of integration: what tasks do they perform, what is the technical knowledge that will be collected in the database and in which context will this tool be used. A thorough familiarization with the work of the engineers is invaluable. All this knowledge gathered during the first phase (T15) will be presented in III and will help understand the objectives of Tanga_DB.

As the future users are now involved in the project, it is important to collect their ideas, comments and wishes regarding the application. It was said that this is the critical aspect of the project: meet the users’ requirements. A tool will become quickly obsolete if it is not adapted, as will be seen from experience in III.B.2. The programmer must therefore have an interactive reasoning with the project sponsors to determine first the high level requirements of the project (directly linked with the objectives) (T13). After a first set of interview was conducted to present the project and list the different methods in use inside
Knowledge management for propulsion systems integration

Matthieu Gonsolin

Each team (See II.B), new meetings are set up to determine the requirements. The different phases and most important aspects of these interviews are:

- Experimentation on the prototype by the user:
The user will handle the tool following his feeling and will be able to comment better the prototype than just having everything presented to him straight away. He might also handle the tool unexpectedly and discover a bug or a missing functionality.
- Presentation of the functionalities he might have missed
- Feedback on the current prototype and the possible ameliorations
- Modify the prototype consequently
- Start a new set of interviews

The prototype presented shall be the same to everyone in each step as people might not have the same opinion. In case of division over a specific function, the programmer should decide along with his tutor which would be the best solution. Some requirements from the user might also be exaggerated and the programmer must keep in mind the feasibility aspect at all times.

From this iterative process, the programmer will be able to establish a conceptual structure of the application and continuously adapt his prototype. The final high level requirements that will emerge from the interviews will be presented in IV.A.1.

At this point, all the potential programming options must be considered and studied. Different software solutions may lead to different functionalities because of their specific limitations or assets. This action (T14) must come along with establishing the requirements as there are strongly related (see IV.A.2).

D. The different steps of programming

Programming an application is nowadays a common type of project and many investigations were conducted to identify a possible structure to conduct these project. In this paper, we will present the V-cycle for software programming, which is quite easy to understand and reflects the main phases leading to a robust application. [10] [11] Even if in the present case, the tool is only developed through existing software (Excel, See IV.A.2) and is therefore only a partial program, the same principles are applicable. This V-cycle structure was applied to the building of Tanga_DB and will be used to justify more into details the division of the tasks presented in II.A (PERT chart).

There are a few parameters to take into account when starting a new software project:

- What is the expected lifetime of the application: months, years?
- Which volume of code can be expected from the main functionalities?
- What are the resources that can be devoted to the project?
- Who are the targeted users of this application?

In this project's context, the application is supposed to last several years since its aim is to capitalize queries over the long term. The application can be considered as small by comparison with classic software examples (a few thousand lines) and the resources are clearly identified as I must finish the project within six months.

A software project can be divided into three main phases (See Figure II.3):

- **Statement of the need** that will lead to the high level specifications
- **Conception** that will lead to class diagrams and structural charts
- **Realization** that will lead to actual prototypes and the final tool

These phases are more or less of importance and considering time cost for 30-35% of the resources are dedicated to the analysis and a typical software project, a good approximation is often that nearly
conception phase, 15-20% to the programming and the remaining 50% to testing. It can be highlighted that testing is often really long and the foremost goal of the V-cycle is to reduce the time required by the last tests’ phases. One might also keep in mind that maintenance is an important part of software development but it comes here downstream of our project.

Looking back at the task division presented on the PERT chart in II.A, we can notice that it corresponds to an exploded view of the V-cycle. Phase I (Analysis of the need and establishment of the specifications) has been detailed in II.B and II.C. It requires interactive discussions with the project sponsors and must be very clear as all the following phases will refer to it. We can point out that assessing the need also involves assessing the existing solutions and their limitations. Phase II, and more precisely T23 and T24, represent the iterative process: Global/detailed conception – Programming – Integration/unit tests. T31 is the equivalent of the systems and acceptance tests that can lead to the users finally handling the application for real.

The V-cycle’s most important aspect is this constant evaluation of the project through several levels of testing. Unit tests enable to debug the lines and make sure that the written program can be compiled and run successfully. Each small function or parts of a module are therefore thoroughly analyzed to validate the detailed conception. The integration tests validate the whole functions and modules as properly working and corresponding to the low-level specifications and structural design of the application. Systems tests verify the accordence with the high level specifications. Finally, the last level of tests must check that the application successfully fulfils the needs expressed in the beginning (See II.E). These multiple layers of testing ensure that errors or deviations from the specifications are detected as soon as possible. For example, the sooner a function is identified not to meet the requirements, the less modification it will generate. To repeat the main ideas, the V-cycle is based on two important principles:

- Continuous interaction with the users at every level
- Development by anticipation: constantly test and adapt the project to avoid large problems at the end

There are different ways of programming. We will see in IV.B the advantages of the object-oriented programming method that will be used to develop Tanga_DB. This will enable us to program separately small modules or functions on a distinct prototype and connect them once they are operational. The programming is then more robust and quicker. Several high-level prototypes will be sent for review to the users (T23_2 /Integration tests).
E. Commissioning

The final phase of the project is the commissioning of the database, which is in fact divided in two tasks (It relates to the tasks T32 and T33 in the PERT chart).

If we go back to the V-cycle presented in II.D, we can see that the last step is the acceptance tests. They can only be performed once the user is actually handling the application and have several purposes:
- Validate that the delivered version of the application is in accordance with the specifications
- Validate the actual performance of the application and its contribution to the engineers’ work
- To a lesser extent, review the mentioned specifications and return to the programming phase for larger structural or functional changes

As was seen earlier, the purpose of the V-cycle is to reduce as much as possible this acceptance phase to a simple positive validation of the work performed. As feedback is constantly provided through the different test phases, a great deal of time and work can be saved by avoiding large changes this late in the development process.

The other step is to fill the database. Once the application has been completely validated by the sponsors, the users start to enter new answers requested to capitalize in-service and production issues. The database remains however relatively empty and the aim is also to capitalize as much experience from the past as possible. Several questions can be raised as it might be difficult to access years of experience that were not capitalized:
- Which is the “right” knowledge to consider?
- To which extent do we want to or can we go into the past?
- Where to get the knowledge?
- What resources shall be used to perform this capitalization task?

This questions lead to more and we will try to answers some of them in V. Even if filling the database or not does not impact the structure of the application, it has a strong impact on its short-term performance: the fewer the problems capitalized, the less support the database provides for new incoming questions.

As the methodology has been presented, we will now focus on understanding the knowledge that will be capitalized in the database, that is to say: in-service and production daily queries regarding propulsion systems integration for Airbus aircraft.
III. Theory and background

A. Technical information

This part presents some valuable information under several categories to fulfill the following objectives:
- Understand better the material behind the database
- Detail the need for a capitalization tool and its objectives
- Justify the choice of information stored in Tanga_DB (See IV.C.1)

First, the complexity of the aircraft engine is described, justifying that issues will inevitably occur on in-service aircraft. Moreover, as the engine is designed separately from the aircraft, it is mounted on the plane and therefore interacts in many ways with it. Both these aspects, interaction and issues, lead to many activities regarding propulsion system integration and Airbus, through its processes and its internal organization, tries to perform all of them in the best way. These activities require extensive communication with the manufacturers and their final aim is to support the customer: the airlines companies.

1. The aircraft engine: a sophisticated system

1.1 General description

The primary purpose of an airbreathing jet engine is to deliver enough thrust to balance the drag acting on an aircraft. This is done by accelerating a stream of air through the different stages of the engine; the reaction force opposed to the high velocity jet flowing out of the engine is the desired thrust. Air is inhaled through the intake and the fan blades where it separates in two: the primary flow and the secondary flow. The secondary or bypass flow is guided straight out of the engine, producing almost 80% of the total thrust. We will see in later that the secondary flow also plays a role for the thrust reverser system. The primary flow goes through several stages of compressors: the Low Pressure Compressor (LPC also called booster) and the High Pressure Compressor (HPC). The air enters then the combustion chamber, where it is mixed with fuel and ignited. The turbines then extract a torque (mechanical energy) from the kinetic energy of the burned gases in order to drive the fan and compressors. The rest of the energy will be converted into thrust as the expanded gases are expelled at a high velocity through the nozzle (See 1.3).
As seen above, one of the fan module’s purposes is to divide the flow and play the role of first stage of compression. It also enables the mount and nacelle attachments (See 1.3), containment of foreign objects entering the engine and attenuation of the noise. Upstream of the fan, on the same shaft, is the spinner cone, composed of two parts. The spinner front cone is an aluminium hollow cone-shaped structure with an anti-icing purpose. The rear cone features several balancing screws necessary for static balancing after a blade replacement or trim balancing in case for example of a too high level of vibrations. The fan blades, nowadays featuring an advanced combination of materials with outstanding mechanical properties, are usually shrouded (to damp oscillations) and fitted on a disk with spacers to limit the radial displacement.

The fan module is housed in the fan frame on which the forward mount will be bolted. Also fitted on the fan frame is the Outlet Guide Vane (OGV) which directs smoothly the secondary airflow between the fan and the outlet. It ensures thrust efficiency and noise reduction.

Commercial aircraft jet engines generally feature two rotating shafts: one linked to the LPT and driving the fan module and the LPC; and one linked to the HPT and driving the HPC only. The accessory gearbox which provides energy to the aircraft and drives the engine accessories gets its energy from the HPC. The shafts are both held by bearings, which are housed in dry sump cavities on the frames. One frame was seen to house the fan module; the other frame surrounds the LPT and enables the aft mount attachment. The rigidity of the structure is ensured with a minimized length between the two frames.

The combustion chamber’s geometry (annular chamber, swirl nozzles and liners) plays an important role. It determines the handling of the flame; ensures an efficient mixing of the fuel, a uniform combustion pattern and low thermal stresses. The fuel consumption and therefore the gas emissions are also directly impacted.

1.2 Systems

The main parts of the engine and their functionality were presented in 1.1. They are the parts that students examine in propulsion courses; they are the parts that come to mind when generally talking about engines. In order for it to work, an engine however needs many more systems, less known by non-specialists, but which makes of an engine the complex entity that it is. They are all the more important as they also impact
aircraft operation: e.g. bleed systems, hydraulic pump or Integrated Drive Generator (IDG). When designing or maintaining an aircraft engine, engineers have to take every part into account, all the way down to a single nut.

Most of the problems occurring on an in-service engine come from unexpected “minor” systems, but the consequences on the integrity of the engine in the event of failure can sometimes be very severe. This phenomenon is even more exacerbated by the straining productivity constraints which demand to always produce cheaper. This leads to more frequent changes in production among the suppliers with potential significant repercussions over quality. The opening to competition on the aftermarket of spare parts, especially in the USA where the anti-monopoly laws are very strict, entails the emergence of engine parts not certified by the original manufacturer, also leading to a cutback in quality. The engineering support for in-service aircraft is then all the more critical to ensure safety (see III.B.3).

As an engine contains several thousands of parts, only the main systems or the relevant units will be presented in this paper. The complexity of a jet propulsion engine will however be already very clear even with this level of details.

Every picture showing the different systems of the engine is taken from the CFM56-5B engine (Figure III.3) produced by the consortium CFM (Introduced in III.D), which is the most sold engine for commercial aircraft today and equips several families of planes. On other engines, the configurations, the spatial distribution of the parts and the architecture can be slightly different. The functions of most systems remain nonetheless the same. [12]

1.2.1 FADEC system and controls

FADEC stands for Full Authority Digital Engine Control. This system enables the engine to be operated from the aircraft by the pilot through different input commands such as the throttle lever. During a flight, the operating point of the engine is constantly displaced, even slightly to ensure maximum efficiency at all times; and as the atmosphere conditions (among else) change, so must the engine parameters. The FADEC has complete control over the engine: power management; starting, shutdown and ignition; fuel, oil temperature; active clearance, variable geometry and thrust reverser. The FADEC is therefore a calculator which performs all the necessary computing to obtain the proper parameters’ values and
command the actuators of the engine to run at maximum efficiency for a given operating point. The FADEC system also provides the aircraft with output data so that the pilot in the cockpit can monitor the engine or report maintenance faults or required troubleshooting. The control functions listed previously can be detailed through some more perceivable tasks:

- Control the fuel injection and set the safe boundary values for the shaft’s rotational speeds
- Control the engine’s start sequence and limit the EGT
- Optimize the operating point through compressor airflow and turbine clearance controls
- Control over the thrust reverser
- Overlook the IDG cooling fuel recirculation to the aircraft tank

We can now look at the physical inner structure of the FADEC system. It comprises two main units: the Engine Control Unit (ECU), which used to be called Electronic Engine Controller (EEC) and is the brain of the system; and the Hydro-Mechanical Unit (HMU), which is the muscle of the system and drives the valves and actuators. Their locations are shown in Figure III.4 and Figure III.9.

Several sensors can be added to those components to complete the FADEC system. The ECU gathers the sensors’ data, performs the calculations and interacts with the aircraft while the HMU converts the ECU’s electric commands into hydraulic pressures for the valves and actuators. There are different types of sensors distributed all over the engine: speed sensors, thermocouples, pressure sensors, resistive thermal devices (RTD) and vibration sensors.

The ECU is a sophisticated electrical system which requires cooling and isolation from vibrations. It features around 15 electrical connectors, each with a specific pattern: power supply, information from the sensors, connection to the aircraft and so on and so forth. With so much reliance of the engine upon automation, the ECU features identical and dissociated channels, providing therefore all the engine functions even with one channel’s failure. It is actually the main disadvantage of using a FADEC as there are no possibilities to manually override the system; meaning that if the FADEC fails for whatever reason, the engine fails as well with no restart possible.

The complete electronic functioning of the FADEC will not be described here because of its complexity and because, as it will be seen in III.C, FADEC related activities are out of the scope of our capitalization tool.

### 1.2.2 Starting and ignition system

A great number of starting systems have been used over the years as the technology developed piecemeal, for example: cartridge starters, compressed air systems, direct cranking and so on and so forth. The purpose of the starting and ignition system is obviously to initiate the rotation of the turbines and compressors and start the combustion up to a point where the cycle is self-sustained and the engine runs autonomously. [13]

On the CFM56-5B, the starting system is controlled either manually or by the FADEC and consists mainly in a pneumatic starter, linked with the LPT’s shaft, and a Starter Air Valve (SAV). The SAV filters the flow entering the pneumatic starter via a duct to adjust the torque (mechanical energy) delivered by the starter. Following the starting sequence orders coming from the ECU, the SAV opens, hence increasing the shaft’s rotational speed; while the igniters are energized and deliver the necessary fuel to initiate the combustion. At 50% of the maximum speed, the engine is considered operative; the SAV is therefore closed and the igniters are de-energized. Figure III.4 shows the above-mentioned components and their location on the CFM56-5B engine.
1.2.3 Air system

1.1 highlighted the airflow path through the engine as the key to deliver thrust. However, air is used for many other purposes in the engine:
- Bleed the flow to supply the aircraft with air
- Cooling, damping and bearing forces balancing
- Variable geometry influencing the airflow
- Clearance control for efficiency purposes (related to cooling)

An efficient air system has a substantial impact on the engine’s performance as the required thrust will be obtained with a lower amount of fuel: enhanced specific fuel consumption leading to lower operating costs and lower environmental impact; and lower EGT leading to an increased life of the engine. As the air system comprises many different systems, only two of them will be described here: the Variable Bleed Valve (VBV), part of the variable geometry controls at the compressor stage’s level; and the High Pressure Turbine Clearance Control valve (HPTCC). We can therefore highlight the intricate network of ducts and pipes around the engine necessary to convey air between all the different fore-mentioned systems.

The main purpose of the variable geometry control system is to ensure a satisfactory performance of the compressor regardless of the operating conditions. When running at low speed, the airflow going through the LPC is larger than what the HPC can handle. In order to prevent the downstream blades from stalling, the VBV’s are installed circularly around the main stream, just after the LPC, and open at low speed to discharge the HPC. On a basic working at high speed, the VBV’s are generally closed. During transitional phases (acceleration or deceleration) or particular conditions, the handling of the system can however be more complicated. For example, the valves can be opened in particular icing conditions to prevent some ice particles to cause damage to the HPC. Figure III.5 shows the ring-shaped structure supporting the twelve valves on the CFM56-5B.
The HPTCC system implies cooling the HPT shroud support structure to reduce the clearance between the case and the rotor to decrease the maximum attained EGT and gain in turbine performance. [14] When the engine is running, hot gases are breathed through the gap between the tip of the blades of the HPT and the casing. Very high temperatures (EGT) can be experienced at this stage, in particular during a sudden increase in power, which can lead to thermal extension of the case and the blade. As the case usually expands faster, the clearance tends to increase, deteriorating the performance of the turbine. In order to limit this phenomenon, cooling air circulates inside the case, limiting the expansion and therefore keeping a safe low clearance margin. This cooling is done through the HPTCC valve. It comprises dual butterfly valves driven by the same actuator, and the associated manifolds. Air is bled from the compressor flow at different stages depending on the flight phase. The flows are then joined before going through the case. On the CFM56-5B, air is bled from the 4th and 9th stage of the HPC.

**1.2.4 Fuel and oil system**

Apart from the air, other fluids are conveyed all around the engine. The fuel pumped into the engine is not only used for combustion but also for oil cooling and powering some of the actuators such as the HPTCC valve. Oil is used to lubricate and/or cool the gears and bearings; and is used in the hydraulic actuators.

The main components of the oil system, apart from all the pipes, are showed in Figure III.7. The system is self contained and might be divided in three sub-circuits: supply, scavenge and venting. The oil is pumped from the tank into the lubrication unit through the anti-siphon device, which prevents the tank to be emptied at shutdown. The oil is then distributed for lubrication and returns to the other side of the lubrication unit through the master chip detector, which is installed on the lubrication unit and allows controlling potential magnetic contamination of the oil. Finally the oil is carried back to the tank through the oil/fuel exchanger.
Each of these components is however a sophisticated system in itself. Figure III.8 shows the main components of the oil tank. As it was seen, the oil inlet tube comes from the exchanger while the outlet tube goes to the lubrication unit. The other ports allow refilling the tank or in case of the drain plug, emptying it. We could however consider now the mounts on the oil tank and detail this system and again on an even deeper level of details. This example highlights the different levels of complexity that are somehow hid behind the main systems and that engineers might need to go down to depending on the issue.

The fuel system is even more complicated than the oil system and will not be detailed. The different purposes of the fuel distribution system are to supply the combustion chamber with clean fuel, deliver clean and ice-free fuel to the various actuators on the engine and to cool the oil through the heat exchanger. The various components of this system are showed on Figure III.9. One can spot the HMU commanding the various servomotors, valves or other actuators or the engine.
1.3 Nacelle

The engine is not mounted directly on the airplane but is hold in a tubular shaped structure called the nacelle. Even though they are fully connected, the engine and the nacelle are usually designed by distinct companies. Comprising four main different parts, this cover serves several purposes:

- **Protection**: The nacelle must prevent any kind of direct damage to the surface of the engine while still allowing the required accesses to the equipment for maintenance.

- **Aerodynamics**: The nacelle must refine the airflow going through the engine to ameliorate the engine performance, and around the engine to minimize drag.

- **Connections**: The nacelle ensures the air, fluids (fuel, oil and water) and electrical connections with the aircraft.

The engine is attached to the pylon, generally located under the wing, by a pair of mounts placed aft and forward of the core section. The forward mount is mainly designed to be able to bear the lateral and vertical loads acting upon the nacelle. The purpose of the aft mount is to restrain the movements of the whole structure in almost every direction. The displacements along the forward-aft direction are prevented by the combination of both mounts. Structurally, the main strut has to withstand all the loads transmitted through the mounts.
As was seen in 1.1, the inlet cowl imposes satisfying entry conditions for the airflow to ensure that the operating point of the engine remains in the required range in every phase of the flight (idle, take-off, cruise, etc). Another function of the cowl is to prevent the formation of ice at the front of the engine. However, the outermost surface of the cowl is also very important for the aerodynamics of the nacelle. Following an approach close to the thinking used for the wings, the external camber of the inlet cowl must be designed to avert high local velocities on the contour. The inner surface of the nacelle parts can be sheathed with different materials, depending on the elements they cover, in order to absorb heat radiations or limit the noise level.

The fan cowl and thrust reverser doors on each side of the engine are maintained together by latches. These doors can be held open to access the engine but the fan cowl doors also features smaller panels to reach directly the starter valve or the oil tank for servicing purpose.

The thrust reverser’s main function is to deflect part of the airflow going through the engine to generate a reaction force reducing or redirecting the thrust, thus providing additional braking during landing. The reverse can however only be used if the plane is in contact with the ground. It improves the safety during landing by shortening the runs and complements the brakes to limit their wear. The thrust reverser has become an essential component in the design of the nacelle as it significantly increases the weight of the system, its reliability, maintenance and design costs.

Located just downstream of the fan unit, the thrust reverser usually ducts or diverts the secondary flow, depending on the configuration. Depending on the technology, the air can either be diverted forward or to the sides by means for example of translating sleeves and cascades or by deploying pivoting doors (2 or 4), leading to different characteristics of the deviated flow. [15]

The exhaust system is divided in two parts whose primary purpose is to accelerate the flow of hot gases exiting the engine so as to provide thrust to the plane. The primary airflow is regulated by going through the annular passage between the nozzle and the centerbody (which can be seen on Figure III.1).

Like the engine, numerous systems have to be included to these main parts to complete the nacelle. There are the hydraulic system, the bleed air system, the drive generator, the actuators and controls for the doors and others. One of the important problems ensued is the fitting of all the connections (fuel lines, electrical cables, control equipments, etc) within the pylon. Another space related issue when designing the nacelle is to minimize its volume while keeping maintenance accesses and still hold the entire appliance.
An aircraft engine involves many more systems but it would take too long a time to describe them all. It is because of this evident complexity that the integration and monitoring activities are as critical as they ensure the proper functioning of the engine and the interfaced avionic systems; and provides a continuous improvement of the reliability of the engine and the in-flight safety level.

2. Technical issues

Given its complexity as a system, an aircraft engine may encounter many problems during its conception, its production and life in service. An engine is designed for specific conditions that may not be satisfied in the future. For example, the automatic start of an engine for a plane based in the Lhasa Airport might experience difficulties related to extreme cold weather conditions, on account that the anti-icing system of the intake had suffered a minor damage on a previous flight, which had been troubleshooted at the time not to impact the reliability on the engine. This just an example of very peculiar situations that the engineers could not anticipate and which illustrates that the engine is prone to technical issues. Below is a list of the main systems or situations that usually face the most problems:

- Propulsion system development tests
- Engine tolerance to icing threat
- Engine vibration
- Oil, fuel and starting systems
- Engine load transfer (cowl load sharing, pylon attachment)
- Fire detectors installation
- Electrical generation system
- Bleed and drainage systems
- Nacelle seals and paint process
- Cowl opening systems, access doors and latches
- Thrust reverser system

To illustrate properly typical issues that can be found on an engine, we give here two examples: one on an in-service engine, the other during production. The solutions to these events will be given as an introduction to the activities of propulsion system integration that will be seen in 3.

2.1 Starter failure

Confidential

2.2 Engine vibrations in climb

Confidential

B. Work processes

The combination engine-nacelle is a sophisticated system in itself “independent” of the plane (See III.A). The aircraft manufacturers do in fact, not design any part of the engine themselves. On the other hand, considerable work is done by the aircraft manufacturer regarding the integration of the propulsion system onto the plane.
Though the main feature of an engine is the performance in terms of thrust, other topics arise when integrating it, such as the management of the aircraft’s energy demands or the weight and drag optimization. Figure III.16 shows different aspects of the engine/aircraft interaction. We will see in III.C how the domains were organized inside Airbus to address all these aspects.

![Figure III.16, Interaction engine / aircraft](image)

As our aim here is to understand the database material, this paper will only focus on the system integration activities, which have been divided into several processes. [18] [19] Figure III.17 is not exhaustive as several secondary activities are not displayed, but the activities of interest that will be capitalized in Tanga_DB are highlighted in dark blue. One can notice the three different categories of activities: Continuous airworthiness and development, Customer service and Continuous support, which we are now going to detail.

![Figure III.17, Integration activities](image)

**1. Continuous airworthiness and development**

Although they will not be capitalized in the database, it is interesting to understand better integration activities to detail quickly some continuous airworthiness and development activities. Development activities (Flow) address the evolutions of the engine: a new design for one part due to costs reduction in
production for example. Continuous airworthiness activities ensure the safety of the plane and its agreement with every authority’s publications.

An ISRO is used to report to the authorities (Office of Airworthiness) all cases where a system loss or significant malfunction happens at a time where system operation is essential and where backups do not perform satisfactorily. As the aircraft manufacturer, Airbus is part of the solving process.

A RFW is an incentive to further investigate a recurrent or critical problem found on in-service aircraft in the scope of improving the reliability of airplanes and reducing maintenance burden, number of delays and cancellations and associated costs. They may lead to opening a modification process (MOD) on some component.

A MISP is an in-service issue classified as Major because the impacts in terms of airworthiness, cost, performance or passenger perception are considered as significant. A MISP is in fact a RFW with a high priority that has to go through a specific process. The main milestones of the process are to deliver a root cause analysis, a list of potential solutions and a design reply along with the right certification file if a modification is required. As always, a strong coordination with the engine or nacelle manufacturer is necessary.

A MOD is the process to develop and introduce a modification of any type in production and also sometimes to implement it on in-service aircraft. A MOD process can be launched for several reasons:
- To treat obsolescence: material, part, etc
- To improve a design following recurring in-service issues
- For economical purposes

2. Continuous support

Even though as many problems as possible are addressed during the development phase of the engine, there always remain unexpected issues that occur in particular during the production and assembly phase. This is referred to as Continuous Support. PSI engineers can be tasked with troubleshooting these events during the installation of the powerplant system (PPS) in the Final Assembly Line (FAL) and ground and flight tests preceding the delivery of the aircraft. They guarantee at all times the coordination with the engine and nacelle suppliers.

Many of these problems occurring prior to Hand Over to the company are solved directly by the production team (MAP) with the help of existent documentation. However, some more complicated issues can lead to a request to PSI Engineers and named Production Queries. These questions can cover a very wide range of problems (See examples below). The role of the PSI engineers is to examine the issue through the maintenance reports, the event data and the related lab investigation if performed, and provide the MAP with recommendations or corrective actions. The timeline for an answer is always relatively short as any problem in FAL can generate a delay on a delivery, which has to be avoided as much as possible. The technical issue on the engine vibrations presented in III.A.2 was a Production Query. Here are some more examples of typical FAL questions or issues that occurred during my internship and were later filled in the database:
- Slight overlap of a redundant hole of the hydraulic pump gasket over the gearbox
- Hydraulic supply line coupling with thread fretting at the pylon interface causing a heavy leak
- Discrepancy between the VBV command and its position
- Failure of the LPC bleed master actuator channels A/B and slightly high EGT
A concession is a signed authorization to use and deliver a product that presents unintentional deviations from the certified configuration caused during the manufacturing process. The concession ensures legally that these deviations do not impact the safety and performance of the product. They will not be capitalized in Tanga_DB.

3. Customer service

The PSI engineers perform the same kind of troubleshooting activities for the aircraft already flying: Airline Routine Question and Technical Adaptation. As was seen in I.A, given the life of an airplane, these activities of Customer service represent an important work for the aircraft manufacturer. These problems usually involve an Aircraft on Ground (AOG). This is a highly sensitive situation where the technical problem is considered serious enough to prevent the plane from returning in the air. Time is critical in these cases as the plane must be restored back into service as soon as possible.

A TA is an approved document under the authority of EASA Design Organization Approval and edited by the aircraft manufacturer confirming that the airline is allowed to deviate from the certified aircraft configuration or a maintenance procedure provided that it is still compliant with the applicable certification basis. In other terms, it ensures that the “damaged” configuration or modified maintenance task do not jeopardize the safety of the plane, which is therefore given a legal permission to fly. Some actions can be required from the airline prior to the release of a TA, which might also be applicable only for a limited number of flight cycles or hours. TAs are the equivalent of concessions for in-service aircraft.

The ARQ or In-Service Daily Queries are treated the same way as Production queries. Most of the problems are filtered by the customer support domain, which sends to the PSI engineers only the trickier issues. Their action is similar: troubleshooting proposals and associated corrective actions. Here are some examples of ARQ highlighting again the wide range of questions that PSI engineers can face:

- Beneficial impacts of ripple dampers removal and tubes re-routing on the hydraulic pump
- Missing coating on anti-ice duct (TA)
- Adaptation of a maintenance task regarding the removal of the spigot ball joint
- How to distinguish between several types of intermixes?

The time allotted to solving an ARQ is usually around 36 hours. As an AOG problem is even more critical, the average time for a response should be less than 3 hours after receipt of the question. Therefore, the specialist usually enters a bona fide race against the clock to find the adapted solution. In those conditions, searching through the existing documentation to see if an identical or even remotely similar problem has already been treated in the past can be of an appreciated use. A simple capitalization tool can therefore be an asset saving time and directing the specialist towards the right interlocutor or written material.

C. Company organizational structure

Having understood the different activities that ensue from the integration of the complex system that is the aircraft engine on the plane, we take a look at the inner organization that Airbus put in place to address these activities.
1. Coordination between the airlines and manufacturers

The direct interlocutor for the airlines in Airbus is naturally the customer service; with the acronym SEE for the domain in charge of engineering support and SEEE the sub-domain responsible for the powerplant. SEE is the focal point for any technical request from the customers but also from internal services such as the repair station. Although they handle most of the queries, on a regular basis, a more thorough analysis is required from the Centre of Competence (See Figure III.20). The answer from the CoC engineers will then be reviewed and adapted by SEE before diffusion to the customer. The requests sometimes get more complicated as the achievement of the objectives involves a third party, for example the engine and nacelle manufacturers. Even if the target date is always adapted given the customer requirements, the possible involvement of a third party and the different levels of investigation required from the CoC; it was seen that time is always a critical issue and all requests have to be solved within the targeted timeframe.

2. Centre of Competence powerplant

2.1 General structure

Within the complex structure of Airbus, a department has been dedicated to powerplant systems (acronym is EP which stands for “Engineering/Propulsion”). The overall function of EP is to handle all the activities regarding the turbine engines: propulsion system and auxiliary power unit system (APU). EE manages the safety, environmental impact, cost and maintenance issues related to an optimal design by the suppliers and integration by Airbus of the engines. As much of the noise surrounding the aircraft is coming from the engines, EP is also responsible for the acoustics. Figure III.18 shows the inner sectionalizing of the domain. [21]

![Figure III.18, Organization structure of EP](image)

The EPD domain is responsible for many different activities such as environmental considerations for engines, knowledge management and feedback maturity (EPDX, see 3.), engineering quality and management processes.

The EPA domain is in charge of every issue related to noise, not only for the engine but also for the aircraft. Involving acoustics simulations, noise measurements and certifications, the activities of EPA are critical today as noise is an important part of the aircraft performance and the airport regulations are getting stricter every day.

The EPV domain is in charge of the APU. Installed in the fuselage tailcone, the APU provides the aircraft with electricity, pneumatic power mainly when the engines are shut down (air conditioning and main...
The EPG domain handles the engine performance mainly related to thrust and specific fuel consumption measurements. The thrust available on a plane is important to balance the drag and the fuel consumption is a key driver of the operating costs. Using advanced simulations tools and test rigs, EPG ensures the aircraft performance prior to delivery and monitors the in-service deteriorations.

Finally, the EPT domain is responsible for the integration of the engine and nacelle on the plane. They are therefore in charge of the different activities that were detailed in III.B.

### 2.2 Engine and nacelle integration: EPT

The EPT domain is responsible for the integration of the engine and is divided in three sub-domains. EPT1 is in charge of the airworthiness and fire prevention, they contribute to establishing the certifications required by the airlines to be legally and safely allowed to fly with their specific engines. EPT2 is in charge of the engine control integration. In other words, they connect the FADEC system (as described in III.A.1) to the aircraft. Finally, EPT3 ensures the engine and nacelle systems integration along with thermal analyses.

![Figure III.19, Organization structure of EPT](image)

The engine and nacelle integration department is divided in sections, each being in charge of a program. On Figure III.13, the SA program represents the A320 family, the LR represents the A330/A340 and the WB represents the A300/A310. Among each section, there usually are one team per motorization, of two specialists responsible respectively for the engine and the nacelle. My local supervisor Mr. Alexandre Armand is the group leader for the A320 program. All of the technical examples related throughout this paper (Engine description and technical issues) are therefore taken from the single-aisle family issues. We know however that one of Tanga_DB’s purposes is to share the experience among these different programs. We can finally notice that the A320neo and A350 XWVB are still under development but will need to be added to the database in a close future.

Many other domains are related to engine integration within Airbus and interact with EPT engineers (See Figure III.20). We can name the Structure domain (ESYI) in charge of calculations, the Mounts domain (ESYT) and others. As was seen from the work processes in III.B, the EPT domain is more in charge of the systems and rather plays the role of a coordinator whereas these departments are in charge of the structure and stress or other calculations and rather play a role of specialist. The Program (BSEN) gives the global engineering directive on an aircraft level.
2.3 The role of EPDX

As seen in 2., the EPDX sub-domain is in charge of engineering feedback and operational maturity inside the EP domain. [22] Their primary role is to back-up on a low level the processes established by KM (See I.A.2). To further highlight the objectives of Tanga_DB and its position in the global scope of knowledge management inside Airbus, we can state the main axes of the EPDX activities:

- Easier and faster access to data sources
- Avoid loss of skill experience and know-how
- Allow the recruited newcomers to access the right information in due time
- Improve the traceability and facilitate design decisions
- Gather airlines feedback (Customer focus groups, Symposiums)
- Capitalize on specific needs for EP

They notably have to implement in EP the following tools: ExTra, RISE, Business Search and IKM.

Tanga_DB is a tool developed to capitalize on specific needs (See I.B) only at the EPT3 level. As it was developed, especially in the primary phase of establishing the specifications, EPDX provided valuable experience on how conduct the project, what to expect from the users and the best practices to follow to implement a new application designed for engineers.

D. Customers and suppliers: an intertwined structure

Important information to consider for our capitalization database is also the complexity of the relations between companies. The civil aircraft industry is slightly complicated as so many companies are intertwined. The aircraft, the engine and the nacelle are manufactured separately. The aircraft manufacturer then integrates the combination engine/nacelle onto the plane but the airline get to choose between the motorizations available for the aircraft. Even though the airline is the final customer, the
a aircraft manufacturer is somehow itself a customer of the engine and nacelle makers as the engine market is entirely contained in the planes sells. In the end, each new purchase brings a substantial interplay among companies regarding commercial and technical issues. Coordination is even more complicated for Airbus as each company develops its own network of representatives for series support, offering numerous possibilities to the airlines of troubleshooting when a snag occurs.

The market is mainly dominated by a score of individual manufacturers and the several joint ventures among them. The major actors of the market are the US manufacturers GE Aviation, whose corporate parent is General Electric, and Pratt & Whitney, included in the United Technologies Corporation; and the British company Rolls-Royce. The Chinese are slowly developing domestic engines in a long-term goal of entering this limited international market.

These companies have however joined through several ventures to share their experience but also the development and production costs and the risk in a quite competitive market. They also form joint ventures to be able to provide engines in other ranges of thrust than their own products to increase their market share without auto-concurrency. For economic reasons as well, this tendency of developing partnerships is much likely to persist in the future to equip the next generation of aircraft.

It can be pointed out than Rolls Royce is currently leaving the IAE consortium while selling its share to Pratt & Whitney. The same trend can be seen among nacelle manufacturers which are often part of a bigger corporation producing engines or aircraft. Figure III.22 shows the main companies competing to sell their nacelles. It can also be pointed out that the engine manufacturers somehow contribute to the nacelle design.
In order to illustrate this tangle of companies, we can take the example of the Airbus A320 family. The aircraft itself is declined into four versions A318, A319, A320 and A321, mainly a change in the passenger capacity. This single-aisle plane is the best-selling aircraft designed by Airbus and includes, among other features: the use of weight-saving composites, an optimized wing platform and new “sharklets” wingtip devices. The picture below highlights the number of engine and nacelle manufacturers involved in powering the A320, which are not less than seven:

As shown, the intertwinement of the companies and the organization inside Airbus is fairly complicated. For any problem on an in-service aircraft, a lot of time and resources might be consumed just to communicate between all parties, transfer information, perhaps settle on a technical issue. This aspect of the work is far from being negligible and is one of the reasons a proper capitalization tool would be useful. It may reduce the amount of communication needed between EPT3 and the other actors, with more
knowledge directly available to the PSI specialists. Knowing which manufacturers, which Airbus domain, which engine are related to a specific query will be important and part of the information stored in the database.

E. Knowledge management in EP

1. Lack of adapted tools

This part’s purpose is to underline the important aspect of developing the database as close to the needs of the users as possible so that they would feed involved and actually use the database (See II.B).

To illustrate this, we can take the example of the Basil database. It was launched a few years ago in this same EPT domain to capitalize the same in-service and production queries. Nowadays, very few persons even remember this database and access it is impossible as it cannot be found anywhere. Basil had many drawbacks (developed under Access for example) but the one thing that people remember is that the tool had been thrown on them suddenly. It had been designed without any consultation of the future users and was therefore off from their actual need. It soon turned out that it would not meet the objectives the engineers had hoped for and Basil quickly became obsolete. We will see in V. some solutions to avoid this situation and ensure that Tanga_DB will be used.

Other examples are the different databases or software available in EPT. There is a real gap between the best practices of the KM domain (See I.2) and the actual every day work of the engineers. This lag can be seen from the tools in use, which are not always adapted and from the engineers’ limited perception of how important knowledge management is. Diderot is an archive system for technical notes where documents as old as 20 years can be found. This tool evolved in a new archive system called EDMS. The problem is that the data capitalized in Diderot could not be transferred so that today an engineer must look in both databases to look for documents. Moreover, both archive systems do not have the same structure and search engine. Diderot only gives a reference whereas EDMS gives a direct link to the document. In the end, the fact of having two different systems for the same purpose reduces their efficiency as people gets annoyed by this non-homogeneity.

2. A need for shared knowledge

2.1 In-group existing knowledge

Following the first set of interviews (See II.B and II.C), an inventory of all the means used by the different teams of engineers in EPT3 to capitalize in-service and production queries was established, revealing several important points:
- No common action is in place to share the data among programs
- No long-term capitalization is currently possible with the available “tools”

Each program or person has its own way to store the information that we can separate into two types of capitalization: personal or shared information

2.1.1 Shared data

Each domain has a common archive folder on an Airbus server to store all the documents related to the activity of the domain. The information is divided by program but everyone in the domain can access the information. Following the same organization, a common archive folder is available under Outlook.
However nothing is properly dedicated to the daily queries and the information can be found in many places under several forms:

- **E-mails:** Most of the in-service and production daily queries are treated only by e-mail. Since a lot of correspondence can be exchanged between the FAL or the SEEE domain, the EPT integration domain, the manufacturers and others departments as was seen in III.C.2; this amount of e-mails makes it very difficult to be stored and even more difficult to search through it for valuable information. Another problem is that not so many engineers take the time to copy the relevant e-mails in the shared archive folders.

- **Technical documentation:** Several topics have been capitalized because of their importance or frequency in diverse technical documents: Technical notes, Engineering Coordination Memo (ECM) and Handbooks for example. These documents mainly contain useful information related to the generic issue and not the treatment of a specific query. We can name once again the recurrent problems related to engine vibrations. When treating a query however, these documents may not be the ideal resource as much time can be spent looking through them for valuable information.

- **Excel sheets:** Each program has built its own Excel spreadsheet to try to list the daily queries. Relatively close to one another, they consist in basic tables giving roughly the date, title, author and engine concerned by a query. It may be a good start to capitalize but these tables present two main drawbacks. They are not filled rigorously by the engineers, depending on the program, on the pressure from the group leaders and so on and so forth. The amount of information stored is also very limited as the primary aim of these tables was more to count the queries treated (For budget calculations purposes) than provide consequent knowledge to ease the treatment of future requests.

### 2.1.2 Unshared data

Each individual obviously possesses his own experience as he should keep a trace of all the queries he has treated in his personal mailbox for example. This data is unfortunately not directly shared with the other members of the domain. Some queries may also be treated only by phone. In this case, no trace at all will be kept of the oral exchanges and the experience acquired will only be accessible by communicating with the concerned person.

### 2.2 Losses of information

All of these methods share the same lack as there are not designed to manage knowledge. Capitalization requires a common dedicated and structured effort to gather the information but there also should be available ways to reuse this information, through efficient search engines for examples. The short description of the tools currently in use shows that nothing is organized: there is not much communication between programs and there is little concern to capitalize these questions, hence little personal involvement. The common reflex when facing a new request is to talk to one’s colleague to see if the issue reminds something to someone. It is not an efficient way to find the information and most of it is lost when changes occur among the personnel or just someone is on temporary leave (vacation, sickness…). All the unshared data that represents most of the work is only available to other engineers for a short period of time (as long as the same person fills the position) and no capitalization is done on a long term for Airbus: in-service and production queries are not stored in any of the existing database, they are too punctual to be part of the ExTra process or to justify a Knowledge book (KCP) (See I.A.2). This overview of the situation, when seen in light of the objectives, tends to strengthen the importance of this project. One can also recall that losses of information can be very negative to a large company like Airbus, especially when new programs are developed. All the experience gained through the older programs should be used to produce an even higher standard of quality. Another argument is the importance of
support to the airline and for future sales rally. The airlines know very well that a plane cannot be perfect but in return, they expect a faultless support.

In order to properly understand the objectives of Tanga_DB, a strong understanding of the activities of propulsion integration and the context of the project was required. One cannot build an adapted database without knowing the data that will be collected and its impact on the users. The choice of information stored in the application will be detailed in IV.C.1 and will require all the knowledge gained through III. We can now detail the conception and implementation phases of Tanga_DB.
IV. Implementation

A. Specifications in accordance with the users’ expectations

1. High level requirements

An interactive discussion with the future users of the tool led to the establishment of high level requirements for the application. They represent the first set of specifications that contractually explain the expectations of the user and that should be verified during the acceptance tests (see II.D). These requirements have been classified in four different categories.

1.1 Accessibility and maintainability

The tool must be easily accessible to the users within Airbus. This simple requirement ensures that the engineers do not lose all their good will to fill the database while spending too much time looking for the application. A multi-user access would be preferable but as the number of questions in a year is not very large (see 1.3 Data), the probability of conflict is very low and a single access is conceivable, if not avoidable.

The tool must be as independent as possible from data processing: software versions and updates, hardware changes (migration to another version of Windows for example)… It is important as many codes can encounter heavy changes as computer science improves and the capitalized data in the tool should still be accessible in a decade.

As little maintenance as possible should be necessary as the programmer will not be able to provide it after the end of the project. The programming code should therefore also be fairly accessible to another person than the original programmer, leaving the possibility to add other features in the future. It is also interesting if the tool should be one day extended to capitalize other activities.

1.2 Interface

The interface must be attractive, user-friendly and pleasant. The commands should be limited to the minimum and the use of the tool should be entirely intuitive (no need of training or user experience). The design must be simple and homogeneous to attract the user. The key term really is user-friendly because if not fulfilled, the tool potential lifetime would be drastically reduced as the users would soon grow tired of the application.

As one of the purposes of the application is to share the experience between the different programs, a single table will be considered, listing all types of request (In-service daily queries, TA and production daily queries) for all programs (Single-Aisle, Double-Deck, Long-Range and Wide-Body). The number of interfaces or pages must be limited to the minimum for simplicity reasons and since only one table is considered.

English is the only language used in the tool, as much for the programming as for the forms and the data. As some in-service or production daily queries might be treated in French, a translation will be required
prior to filling the database. It ensures homogeneity to filter the data and ensures that every user will be able to understand the information contained in the application.

1.3 Data

The total volume of requests for all programs has been estimated from experience around 500 new entries a year. The application is capable of supporting this estimated data volume over a period longer than several decades.

The flexibility aspect of the data is very important. This involves the possibility for a user to detail as much as possible a request while keeping the mandatory fields to the sole minimum that the user would like to find when searching through the database.

The application, through the User Guide and its structure, must give clear information to the user on its inner limitations (the maximum number of characters in a field for example).

The tool must foremost protect the data. Capitalizing knowledge involves storing the data without it being altered by time or incorrect handling from the users. A particular attention is given to limit these potential data losses.

As the amount of information relative to a query stored in the database will be limited or the query might refer to specific documents, a way to relate to these additional documents is necessary. It can be in the form of attached documents, hypertext links or simply the address of the document. This requirement is interesting but is not a priority of the application.

1.4 Performance and functionalities

The objective of this tool is to capitalize in-service and production daily queries in order to ease their treatment and lower the time to answer. To achieve this objective, the application must feature three functionalities:
- Filling: Be able to add any new query to the database
- Search: Go through the database to look for queries matching a particular criteria
- Consulting: Access as many information as possible on the relevant query

The tool must adapt to one’s practices: an entry can be filled in one or more times and the search engine should be flexible enough to provide multiple filter options.

The key terms to describe the expectations of the users in terms of performance and functionalities are the following: simplicity, quickness, accuracy and homogeneity. Quickness, accuracy and homogeneity are essential as the aim is to provide support to the treatment of daily queries. The tool must provide the best time saving and find the closest queries related to a search in the database. Simplicity and quickness are important so that the tool will be used and not abandoned quickly because of its complexity.

2. DBMS and spreadsheets

From these high level requirements, a study of the different software options available can be done in order to choose the more adapted support to the programming. We present here the reasoning leading to our choice of using a simple spreadsheet (Microsoft Excel) and a friendly interface via Macros.
A database is primarily a collection of data; organized in a way to support different processes requiring accessing this information. In today’s common language, the term “database” usually associates both the data and the software to handle it. A DBMS (Database Management System) is a software system specifically designed to handle various databases: creating, sorting and modifying data and data structures. It provides the visual interfaces to the users to easily manipulate their information. Many systems have been developed over the years, the most famous being MySQL, Microsoft Access and Oracle. As said, the term “database” can also refer to a simple collection of data such as a spreadsheet. The most common software to handle spreadsheets is Microsoft Excel, available in any company nowadays.

Going into the details of the properties and functioning of a DBMS would not be relevant in this paper. The main functionalities featured by most existing DBMS and their limitations in our case can however be presented to justify the choice that was made to build Tanga_DB on a simple spreadsheet. These functionalities can be divided as follows:

- **Data definition:** Define new data structure, modify or remove existing structures.
- **Data maintenance:** Define new data, modify or remove data from existing structures
- **Data retrieval:** Apply different queries to sort, analyze and extract data from the tables
- **Data control:** Data integrity and protection, system failure, users interface

DBMS are generally very expensive and hard to maintain: they require buying the software’s licenses, they require dedicated server space to store the data and they are subject to many software updates. In Airbus and more precisely in the EP domain, the only already available database system that could have been used was Microsoft Access. However, it appeared from the set of interviews (See II.C) that not so many engineers were be comfortable using Access or another complicated DBMS. We saw in III.E.1 that the Basil database under Access was quickly abandoned. A spreadsheet is much more limited regarding data structures’ handling but enables to store simply large quantities of data.

The main advantage of a simple Excel is its availability and easiness to use. Every engineer knows how to manipulate Excel. In order to ensure its use over the long term, to limit the cost of the project and to develop the easiest application possible, it was decided to program Tanga_DB under Excel. Relating to the high requirements stated before, we can point out the advantages and limitations of using a spreadsheet.

- **Advantages:**
  - Available in Airbus and of common use
  - Easy to use and dependent only upon the Excel version
  - Friendly interface and dynamism possible using embedded macros
  - Can support the data volume specified (1,048,576 rows maximum)
  - Low maintenance, accessible VBA programming to beginners

- **Limitations:**
  - Fixed structure of the table (headers)
  - No multisuer access possible when running macros
  - Limited storable data in a cell (one property) (32,767 characters)
  - Increased processing time with large tables (for searching mainly)
  - No attachments possible, use of hypertext links only
  - More adapted to numbers than to text compared to Access

Apart from the above listed limitations, the Excel solution can meet all the requirements stated in IV.A.1. The PSI engineers’ reluctance to start using a more complex DBMS led to this decision quite early in the project and many specifications were already written considering an Excel based application. The specificities of Excel regarding programming will be presented in IV.B.
3. Low level requirements

As was explained in the description of the phases of the work, an important part of programming an adapted tool comes from an interactive scheme of improvement-presentation-feedback with the future users. Instead of developing entirely the application based on the sole high-level requirements and then present it to the users, a more effective method is to continuously adapt a prototype, which the users should review as often as possible to give their comments and preferences. Following this scheme can save a lot of time as your application will undergo more changes step by step, closer to the users’ expectations rather than proving to be offset from the need late in the programming process, leading to heavy structural changes of the application for example (See V-cycle in II.D). However, a drawback of this method is the possibility to lapse into the trap of having too much feedback, with reverse decisions and contradictory demands among the users. Some personal choices or decisions have to be made by the programmer and the hierarchy when a disagreement develops over one aspect of the application.

Tanga_DB has been developed with a constant supervision from my tutor Mr. Armand and tested several times by the EPT31 team. Two accomplished prototypes have been presented to the EPT31 leaders in order to get their validation. The tool has then been presented to every user for a final set of comments and delivered for commissioning after a few finishing touches and the green light from the hierarchy. Prior to the commissioning of the final tool, the EPT31 team was already asked to use the second prototype in order to start assessing the actual performance of the Tanga_DB database (see V) and back-up the debugging phase (see IV.D.1).

This interactivity with the project sponsors led to another set of requirements on a lower level. Some of them were expressed during the first set of interviews conducted; others are specific to the Excel proposition and arising from the prototypes’ utilization (see II.C). These requirements are described in the following paragraphs and may ease the understanding of the tool and its structure in IV.C.1. Some of the programming choices made to meet these requirements will also be presented in IV.C.2. Being on a low level means that some of those requirements already are solutions to meet the high-level requirements or a detailed aspect of a larger functionality. However they still remain directions from the users, part of the detailed specifications of the application. Following the high-level requirements division of IV.A.1, the low-level requirements are divided amongst the three main functionalities and additional features:

- Filling: Add a new entry in the database, lately referred to as NEW
- Searching: Go through the database to match a specific request, lately referred to as SEARCH
- Consulting: Includes the display of the complete database and the display of a particular item which will be referred to as ALL and DETAILS respectively.
- Other additional features: Protection, User guide…

One can notice that this division matches closely the main functionalities of a DBMS presented in IV.A.2. As the specifications shall be respected by the tool, the following sentences have to read as much as a description of the application as requirements.

3.1 NEW

The NEW function enables a user to enter a new item in the database whenever he just solved or received a query. Using a simple form, NEW displays a limited number of fields to fill in and then create automatically the new line in the database.

For simplicity and quickness reasons, and after a survey conducted within the domain, it was concluded that an entry should not have more than fifteen fields. Above this value, much more users see the task of
filling as a chore and the tool quickly palls. Tanga_DB features fourteen different fields, a trade-off between the amount of time required to fill the form and the amount of knowledge capitalized in the tool. The different selected fields, which are part of the requirements, will be discussed in IV.C.1. However, some of them need to be introduced earlier:

- “Program”: Identifies the type of plane on which the issue occurred (A320, A380...)
- “Origin”: Identifies the service issuing the request (FALs, SEE...)
- “Date”: Identifies the date of the issue (which is usually treated within one or two days)
- “Author”: Identifies the person responsible of the query within EPT3
- “Title”: Clarify the issue: what system, what defect, which airline? and so on and so forth

For accuracy and homogeneity reasons, most of the fields are in the form of locked dropdown menus. This solution imposes to the user to select only a suggestion from the list. Using this feature will ease greatly the search and provide more accurate results as the data will be homogeneous. To highlight this point with an example, one could consider a field “Program”, telling on which aircraft the issue has occurred. For an A320 plane, two different users could write either “A320” or “Single-Aisle” and a search by keyword for “Program” might not give all the associated results.

For simplicity and quickness reasons, a distinction must be made between mandatory fields and optional ones. Some information is regarded as indispensable, in particular when consulting an entry; other fields can provide additional helpful information on the query if the user has more dedication and especially more time to fill the NEW form but do not impact greatly the performance of the tool. Some management of these mandatory fields is required: advising sign (x), impossibility to validate the form with an empty mandatory field (warning message and coloration of the concerned field). It also ensures not to have empty cells in the database which is a critical issue for Excel.

For accuracy and homogeneity reasons, a formalization of the “Title” is suggested (not mandatory). As was seen in III.A.2, the range of technical issues covered by production and in-service daily queries is very wide. Also, given the different “Origin” of the query, the formalism of the question can vary a lot. In order to display a readable list of “Titles” to the user, the necessity of a common format is indisputable. Following the same principle of the credit card codes, the “Title” is divided in three smaller parts, linked together to create the unique “Title” of the new entry. The division will be presented with the tool in IV.C.1.

For simplicity and quickness reasons, relations are established between the different dropdown menus. For example, when choosing a “Program”, the user can only choose among a limited set of “Engines”. It also enables to narrow automatically the SEARCH.

For accuracy and homogeneity reasons, the “Origin” and “Author” fields must be locked dropdown menus but dynamics at the same time. This means that if the user cannot find the right “Origin” or “Author” in the menu, he has the possibility to add a new one to the list. There are two differences with this method than using a plain textbox:

- The list increases the chances that the format indications are respected when creating a new item
- A same “Author” or “Origin” cannot be added two times with a different orthography, as the user will choose in the list first. For example, the database will not have “M.Gonsolin” and “Gonsolin Matthieu” which would represent the same “Author”.

As Excel cannot attach documents to the spreadsheet, the user is able to add up to three hypertext links when creating a new query. If more documents need to be addressed, they can be joined in a common folder with a link pointing to this folder.
3.2 SEARCH

Through the SEARCH function, the user can find any query in the database matching his search criteria. Like NEW, SEARCH uses a single form with locked dropdown menus and a textbox for keywords. The number of results and the relevant queries are displayed at the end of the SEARCH.

For simplicity reasons, the user can fill any or several fields of the form and the program will apply the cumulated filters on the database. Like NEW, the use of locked dropdown menus ensures more accuracy of the search (matching criteria for the corresponding fields). The number of fields is also limited only to the relevant ones. A field to search by “Date” is available, allowing the user to search a specific date or a period of time. The day and month criteria are not mandatory and if empty, uses default values. The textbox “Keywords” can be filled with as many words as desired, separated by blank spaces. The SEARCH function will follow an OR logic to go through the database.

For accuracy reasons, the database is flexible regarding case and punctuation. However, some instructions must be followed: “Date” format and space between keywords.

For quickness reasons, the criteria after a SEARCH are stored and suggested as default values for the next search. This allows the user to change only one keyword for example to restrain his SEARCH and save time. A possibility to reset all the criteria is also available.

3.3 DETAILS

The database is displayed as one table on one Excel sheet. When a NEW query is created, it comes at the top of the table. After a SEARCH, only the filtered queries are displayed. The ALL function allows displaying again the complete database.

For a more pleasant display that fits in the screen, only the most important fields are displayed. This allows having the entire row displayed on a classic screen, therefore limiting the need to scroll left or right on the page. In order to see the complete information of the query, the user must click on the query and open the DETAILS. As the number of entries in the database is very large, a simple use of division keeps the headers and buttons visible when scrolling down the table.

DETAILS is a single form, completely identical to the NEW form and display all the different fields. The values of the fields can be changed to modify a query, or in case of a user filling an entry in more than on time, complete it. A possibility not to save the changes is available to prevent unwanted modifications or typing errors.

The DETAILS form is a more user-friendly display of a query (form instead of a row) and the fields are more readable. For the same reasons, a PRINT option is available that uses a template to print the query’s information as a PDF. He can therefore have an electronic or paper version of the query, easy to read and presenting all the information stored in the database on the topic.

Several details can be opened at the same time to compare different technical issues and see for example, what differed between the solutions implemented. These forms are independent and can be manipulated at wish.
The database can easily be sorted by any of the fields. A SEARCH sorts by default the queries by most recent date. In order to DELETE a query, the user also needs to go through DETAILS and an additional confirmation of the suppression is requested.

3.4 Protection

Protection of the data is one of the most important aspects of the tool as the aim to capitalize information is to carefully store it for a long time without altering it. This includes protection of the data and protection of the application.

As was seen, a direct modification of a query on the spreadsheet is impossible as the sheets are protected. The DETAILS form is an intermediate between the user and the data, offering protection and a better display at the same time.

Expect for the form DETAILS that needs to be opened several times in parallel, the other forms of the application are modal, meaning that the user cannot undertake any other action until the form is closed or validated. This limits the potential errors coming from a misuse as the user is somehow forced to follow the program. For the same reasons, the closing cross in the top right hand corner of any windows has been deactivated so that the user is compelled to use the new built-in buttons.

When closing the application, a confirmation for saving the workbook is requested in order to prevent, in case of an unexpected bug, to save automatically the tool with the fault and jeopardize the integrity of the application. This confirmation can also compensate for unwanted modifications from the user.

When closing the application, the “Home” page is activated so that, at the next opening, the user will see the warning message asking to “Activate the macros before any other actions”. The policies regarding the use of MS Office at Airbus do not allow activating automatically the macros. Therefore the moment where the application is most vulnerable is at the opening when the macros have not been activated yet. In addition to the warning, the tabs are hidden, forcing the user to use the built-in buttons to navigate between sheets. He is then stuck on the “Home” page until he activates the macros, which protects the database.

3.5 Others

The application comes with a User-guide (available in Appendix 2) of one page: clear, short and easily readable that highlights the limits of the tool and best practices. User-guides often annoy the users by their length, complexity or unintelligibility whereas in this case, the aim of the guide is to fuel the need to use the tool while giving simple advice. As one of the top requirements of the application is simplicity, there should not be a need for a long user-guide anyway.

For maintenance purposes and in order to ensure the continued existence of the tool, the programming must be easy to follow by an external person, non-expert with the language. The code must include detailed comments on the actions performed, with for example well-thought-out names of variables and functions. It will enable the EPT team to adapt or correct easily the code in case of a bug or a limiting feature. For efficiency reasons, the programming should be optimized to limit the processing time, moreover since saving time is one of the main issues of Tanga_DB. A small manual providing programming instructions has been written in parallel of the user guide.
Along with the application comes a specific archive folder available to store relevant documents. Hypertext links related to this folder will remain valid as long as the application and archive folder are linked together and therefore have less chances to be obsolete than links pointing to distant documents.

For simplicity and homogeneity reasons, the number of buttons on the application is limited and to each button correspond a pair colour/function.

With all these requirements in mind, high level and low level together, we can now discover the visible interface of ’Tanga_DB’ under Excel and understand better the choices made to meet these requirements. One can also be reminded, as explained in II that the fact of being close to the need of the users is one of the few ways to ensure that the application will be effectively used.

**B. Programming under Excel**

1. Object-oriented programming

Although this paper is not aimed at giving a complete lesson in object models, an important part of the project was the programming of the tool through the Excel application. In order to connect the specifications detailed previously and the final functionalities of the tools, present the different phases of the development, the choices made, the main issues encountered and the solutions or way around featured in the code; some basic knowledge on object-oriented programming and Excel will first be given.

OOP stands for Object-Oriented Programming. It is a specific programming paradigm based on the concept of “objects” interacting with one another as opposed to the conventional methodology featuring a plain list of functions and procedures. These objects are organized into a hierarchy of classes in order to have structured groups of similar objects. Objects have to be seen as some data (their properties) assisted by a set of functions (their methods) that ensures that the data is used properly. Each object is at the same time an instance of the class, possessing the inherited set of properties and methods, and an individual entity featuring its own characteristics. [23]

As objects often reflect real-life entities and to understand better the concept, consider the example of a library database. There could be one object “book” associated to every real book in the library, with its own properties (ex: Title, Author, Status, etc). There could also be an object “library” that would interact with several objects “person” through a method “lend”. The function “lend” of the “library” could be programmed by changing the “book” property Status from “stored” to “lent” and the “person” property Loans changed to Title of the “book”.

The main advantage of the OOP compared to other methodologies is the possibility of easy code reuse and improvement. Because the technology focuses on data through individual objects rather than processes, it is possible to address local problems or add a complementary module without impacting the complete application. This is the reason why OOP is today the dominant programming methodology as the programs became larger and larger but needed to remain manageable. OOP can however become a very complex concept as soon as you start going into the details of the structure and the relations between objects (encapsulation, subtype polymorphism, inheritance, etc). This paper will present only the relevant part of the Excel object model on a high level.

During the past years, many languages have been developed following this OOP. The MS Office suite, including Excel is built on an object model and provides several options for developing applications
interfacing with it. The XML language was the first available but has now been superseded by Visual Basic for Applications (VBA) derived from the original VB. According to Microsoft, “VBA code objects are associated with Excel workbook objects. VBA allows event trapping, customization, and the addition of user-defined functions and commands”. VBA also uses the Component Object Model (COM) standard for Windows-based application to communicate with Excel. The other main interface available is the C API (Application Programming Interface) using the C or C++ languages. It is a direct and faster interface that however comes with two drawbacks: increased complexity and therefore development time, and increased likeliness of crashing errors. As robustness is one of the most important characteristics of the tool, the programming has been performed through the VBA interface, which we can also refer to as Macros.

The particularity of programming a Microsoft Office application using a VBA interface is that Excel (or Word, or Access...) already has its own complex object model. Therefore, the programmer usually manipulates the embedded objects rather than create his own classes. The programming strategy is then slightly different from a common C++ application for example, since the programmer, instead of creating a class or a method to develop a feature of the application, prefers to find and reuse the proper objects and their methods within the build-in model.

2. Excel class diagram

As was seen in IV.A.2, the main difference between a DBMS and a spreadsheet is the absence of data structure. Data in Excel is stored through lines and columns, which is sufficient for our application: a query is stored in a line, each of its properties on a different column.

The Excel class diagram is much extended and it would be pointless to present it all. However, as most of the objects and properties are based on real objects (Worksheet, Cell) or functions available in the menu and commands bars (Font.Color, Find), it is quite easy to understand. Figure X shows a simplified and partial version of this diagram taken from [25].
In order to simplify the understanding of IV.C where the tool and the code will be presented, it is useful to introduce several objects and properties:

- **Application**: Object representing the Excel application currently running
- **Workbook**: Object representing one file opened under application. ThisWorkbook is a specific object relating to the file that contains this code when used in the programming.
- **Worksheet**: Object representing one tab of a Workbook
- **Range**: Object representing a set of cells, continuous or not
- **Range.SpecialCells**: Property of the Range object representing all the cells of this Range of a given type, for example, the visible cells or the empty cells.
- **Userform**: Object representing a dynamic form that can be opened to interact with the user

---

**Figure IV.1, Excel class diagram**

![Excel class diagram](image)

---

In order to simplify the understanding of IV.C where the tool and the code will be presented, it is useful to introduce several objects and properties:

- **Application**: Object representing the Excel application currently running
- **Workbook**: Object representing one file opened under application. ThisWorkbook is a specific object relating to the file that contains this code when used in the programming.
- **Worksheet**: Object representing one tab of a Workbook
- **Range**: Object representing a set of cells, continuous or not
- **Range.SpecialCells**: Property of the Range object representing all the cells of this Range of a given type, for example, the visible cells or the empty cells.
- **Userform**: Object representing a dynamic form that can be opened to interact with the user
One can notice the hierarchy between objects: a Range belongs to a Worksheet, which in turn belongs to a Workbook. The dark blue in the diagram represent objects and the light blue represent properties. One can notice that Worksheet is an object, but Worksheets is a property of Workbook referring all the tabs contained in this Workbook. Range.Font is an object, but Range.Font.Color is a property. Even if the class diagram is pretty straightforward to understand, it has its complexity. We can also give some examples of the Object methods that will be most used in the program:

- **Worksheet.Activate**: Activates the related tab as the user would do by clicking on it
- **Range.Find**: Search for a specific criteria inside the related range
- **Worksheet.Protect**: Protect the related sheet as the function Protect in the Revision menu would

The VBA application is constituted of Worksheets, Userforms and Modules (which contain the main function) that interact with each other whenever an event is triggered, as for example:

- Click on a button
- Double-click on a specific cell
- Opening the Workbook
- Initializing the Userform

The application runs back and forth between the different modules and functions, creating the dynamic interface that the user sees on his screen. More on the functioning of Excel and VBA can be found in [24], [25] and [26].

As seen, the Excel object model being close to reality, it is easy to understand how the program generally works. Programming advances functionalities is on the other hand trickier. With this basic Excel knowledge and remembering the requirements stated earlier, we can now present the application itself.

**C. Developing a user-friendly application**

1. **Presentation of Tanga_DB**

1.1 Choosing the proper fields

The first things to present before the tool itself are the selected fields detailing a query. The visuals of the main pages and forms will then be presented along with the terminology of the application.

This list of fields was determined as part of the requirements. It was seen that only a limited number of fields is acceptable, and that a possibility must be given to detail the query. Some other fields were suggested, but after the prototypes suffered several modifications, a trade-off had to be made between the amount of information capitalized and the quickness of the application (Filling and searching).
Figure IV.2 presents the different fields considered, grouped by categories of information (See III.):

- **Technical information:**
  - **Title (Object and defect):** As was seen earlier, the field Title is divided into several points. The object and defect can infer properly what type of problem is capitalized. As we saw in III.A.2, there exists many types of requests and it seemed pointless to the users to categorize them. The focus is put much more on the systems impacted through Object, Equipment, Engine/Nacelle and ATA.
    - Ex: Fan blade – Thermal fracture, Oil filler door – Opened latches, etc
  - **Equipment:** It might sometimes be redundant information with respect to Object or ATA but this field can categorize the different queries by the system or part related. Equipment can relate either to a system, a sub-system or a part of the engine
    - Ex: Fan cowl door, Starter air valve, etc
  - **ATA:** Every system or part of a plane, phase of flight, etc, belongs to one category of the ATA classifications (wings, mounts, fuel, hydraulics...) used to delimit the sectors of activity within Airbus. For better understanding, this ATA classification is given in Appendix 3.
    - Ex: Fuel, mounts, hydraulics, etc
  - **Question:** Enables to detail the issue and question that needs to be solved
  - **Answer:** Enables to detail the solution to the issue. Contrary to Title, Equipment and ATA that gives general information of the type of issue, Question and Answer give thorough details about the request treated.
  - **Links:** Enables to add relevant documents to the query: data papers, reports, technical notes, etc
  - **Engine/Nacelle:** A distinction can be made between engine related issues and nacelle related issues.

- **Work processes:**
  - **Category:** Classify by type of integration activities: Continuous development (Flow), Customer service and Continuous support.
    - Ex: In-service query (ARQ), TA, etc.
  - **Origin:** Associated with Category, it indicates the original sender of the request
    - SEE, FAL, Internal EPT request, etc
  - **Date:** Indicates the date of reception of the query
Knowledge management for propulsion systems integration

Matthieu Gonsolin

- **Company organization:**
  - **Program:** Indicates the type of plane on which the issue occurred. We saw in III.C that to each program is associated a sub-domain in EPT3.
    - Ex: Single-Aisle, Double-Deck
  - **Engine:** Indicates the type of engine on which the issue occurred. We saw in III.C that to each motorization is associated a team in EPT3.
    - Ex: CFM56-5B, IAE V2500
  - **Author:** Indicates the person that handled the query inside the domain

- **Supplier:**
  - Program and Engine also indicates the supplier or manufacturer involved in solving the issue.

- **Customer:**
  - **Title (MSN – Company):** The first part of the Title indicates the serial number of the plane and the company that related the issue in case of an in-service problem

As the requests can cover a very wide range of technical domains and systems (See III.A.2), a classification of the queries by problem was rejected: a locked menu with a list of potential problems would have been too heavy and not exhaustive. The solution adopted are the “Equipment” or “ATA” fields that give more chances to perform an accurate search if filled. The division of “Title” goes also is this way as the user must try to state every query as a defect of an object.

With only several mandatory fields (See the (x) and grey background on Figure IV.6 and Figure IV.7), enough information can be capitalized in the database to allow a user, when performing a search, to find if a similar query has been treated in the past, when and by whom, and give him a lead to look for additional information. The mandatory fields are also distributed over all the information categories.

Even if the number of field is quite significant, the dropdown menus are nevertheless easy to fill (“Program”, “Engine”, etc) and overall, create a new query is pretty quick and straightforward. The user is still fairly allowed to fill a query his own way (“Adapt to everyone’s practises”, see IV.A.1), the application therefore meets the simplicity and quickness requirements; and sufficient information is available for an accurate search, provided the database is filled correctly.

With all the information given previously in this paper (integration activities, objectives and requirements of the application and Excel features), a visual presentation of the tool can finally be given. All the screen captures in this part show the final version of Tanga_DB at different moment of the filling phase and containing at least real examples that were required to debug the tool and prove its performance (See IV.D). The tool presents two pages: “Home” and “List” with only one table for all types of questions and all programs.

### 1.2 Visual presentation of the structure

Figure IV.3 shows the relations between the Worksheets and the Userforms along with their main functions. They constitute the visual interface of the application with which the users interact. Most of the functionalities have been explained already through the low-level requirements in IV.A.3 so we present here more the visual aspect of the application.
1.2.1 Home

When opening the Workbook, the user arrives on a “Home” page. This page contains a warning telling the user to activate the macros before any actions. The IT policies at Airbus disable the automatic activation of the macros, which must be activated in order to access the functionalities of the tool and run the code. Three buttons are available on this page; corresponding to each of the three main functionalities (see IV.A.1): filling (Button “New”), searching (Button “Search”) and consulting (Button “To List”).
1.2.2 List

The button « To List » enables the user to go directly to the database that is displayed in Figure IV.5. A larger version of this figure is available in Appendix 4. One can notice the order of fields: they are not ordered by category of information but organized with the mandatory fields first and for programming reasons, with the dropdown menus first.

The display in Figure IV.5 is not cut to the right: in order that the complete database would fit in a standard screen, some fields have been hidden (“Question”, “Answer” and “Links”) as they would anyway not be readable in such small cells. The “Details_Item” form that will be seen later in this part makes up for this problem. The Worksheet is fractioned so that the buttons and headers at visible at all times when scrolling down the table. The different activities are divided in colours representing the different budget categories: Continuous development (Flow), Customer service and Continuous support. The “Title” field is highlighted as it is the most valuable information of a query. Some basic instructions are given to guide the user through functionalities. From the near empty database below, one can already distinguish the mandatory fields and the optional ones as some of them are not filled, revealing different practices among the users.

One can also notice the same buttons than the “Home” page: “New”, “Search” and “To Home” to which we can add an “All” button. This button’s function is to reinitialize the display of the complete database: delete the filters after a search and sort by most recent date.
1.2.3 NEW

The button “New” opens the form entitled New_Item to create a new query. As there are dynamic menus, the “Author” and “Origin” fields have a + button enabling to add new entries to the lists. The associated forms are called New_Author and New_Origin respectively. The hypertext links use the embedded form from Excel to navigate through the computer and find the right document. The display of the form was optimized to fit in a standard screen with as much space as possible for the large fields such as “Question” and “Answer”. The order of the fields on the form follows the order in the database that was explained earlier.

Figure IV.6, New_Item form
One can notice the mandatory fields on the form. When validating the form before filling all of them, a warning opens and the empty fields are highlighted. When all the mandatory fields are filled, the user can validate the form and a new entry is created on top of the database.

![Figure IV.7, Management of the mandatory fields](image)

1.2.4 SEARCH

The button “Search” opens the form called “Search_Item” enabling the user to filter the database to find queries matching different criterion. Every time a search is performed, the criteria are kept in memory. The button “Clear” can reset all of them. One can notice that only the dropdown menus have their own criterion as the “Keywords” field will apply to all the other fields (“Equipment”, “Title”, “Question” and “Answer”). Figure IV.8 presents an example of a SEARCH: look for queries treated by Armand.A on the CFM56-5B engine for the A320.
1.2.5 Sort

As can be read in Figure IV.5, a sorting of the database is possible by double-clicking on one of the headers. Figure IV.9 shows an example where the user has sorted the database by “Author”.

1.2.6 DETAILS

As the complete table is not displayed and some “Question” or “Answer” fields can be quite long (within the limit seen in IV.A.2), a specific form for consulting was added to the requirements. This form called “Details_Item” is opened by double-clicking on any of the fields of the desired query. As many queries as
possible can be opened at the same time to compare them. The “Details_Item” form is similar to the “New_Item” form as it can be used for consulting but also to modify a query. To protect the data, it is not possible to “Delete” directly a query from the database but only through its “Details” form and yet another confirmation is still required. The user can access the different links of the query (if any) by clicking on the redirecting arrows. As the fields can remain small compared to the amount of text stored in it, a “Print” function is available, using a template to save a PDF version on the computer. Figure IV.10 gives an example of “Details” by opening the third result of the previous “Search” example. The print preview of the query using the template is available in Appendix 5.

![Figure IV.10, Details_Item form](image)

*Figure IV.10, Details_Item form*
When closing the workbook, it returns automatically to the “Home” page so that the next user sees the macro warning first.

Overall, this tool is has been built to be as user-friendly as possible while fulfilling the global objectives. The number of functionalities is adapted to the need and optimized to save time for the user.

### 1.3 Clarification of the main features

Some additional information must be given regarding Tanga_DB to see that all the requirements specified in IV.A are fulfilled.

The tool is accessible on the common server in a dedicated folder with a possibility for the user to create a shortcut on his desktop and access it directly. Having a visual icon of the tool may also help the user to remember that he has to fill the database. Given the Excel solution with the extensive use of macros, it was seen in IV.A.2 that a multi-user access is not possible. However Excel opens a warning when the application is already used by someone else and another message when the tool is finally available to be used. Therefore, as the probability of conflicts is very low and the time spent by the user on a search or a new query is very short, this aspect will not be restrictive.

Also because of the macros, the global Workbook cannot be protected as the macros will need access to the data when running. However, many protection measures have been taken to ensure that the application will work properly and that the data will not be compromised:

- **Locked cells:** Both pages are locked but enable the interactivity with the user (buttons and clicks). No cell can however be modified directly: the query fields can be modified through the form “Details_Item” as was seen earlier, the other cells cannot be modified at all.
- **Modal forms:** Except for “Details_Item”, the forms are modal meaning that no other action can be taken until the form has been closed or validated.
- **Hidden tabs:** To prevent the user from changing the names of the pages or adding new pages to application, the tabs are hidden and dynamic buttons are used to navigate between them.
- **“Home” page:** When closing the workbook, the “Home” page is activated so that the next user will find the macro warning and, as the tabs are hidden, will not be able to access the database until they are activated. Activating the macros also initialize the application and reset the display: zoom property, deletion of existing filters and protection of the pages.

The “Search” function is divided into two sets of criteria: the dropdown menus that filter a specific field and the “Keywords” textbox that will go through the remaining fields. Doing so will save a lot of time when the database will be filled, after one or two years. The “Sort” function can apply to any of the visible fields in the database but a combine sorting (by X then by X) is not possible. Some document's formats are not supported by Excel. Hypertext links cannot be created toward these documents so the links should refer to the parent folder instead.

The application has been optimized as much as possible (number of mandatory fields, number of pages, buttons, management of the forms, code for the search function and so on and so forth), leading to a user-friendly, simple and quick tool capitalizing in-service and production daily queries.
2. Programming choices and highlights of the code

As the programming phase represents a large share of the work performed on the project, some of the functionalities presented in 1. and through the low level requirements in IV.A.3 deserve to be looked upon more closely. Several features that could not be realized on the final tool but that were investigated are also discussed. The small code sections detailed in the following paragraphs represents either the more important functionalities of the tool or the trickiest functions in terms of programming and can be very useful to anyone intending either to create an advanced database with VBA or understand the other side of Tanga_DB. Like in the previous parts, the three main functionalities and additional features will be distinguished.

2.1 NEW

When opening a Userform (NEW for example), the dropdown menus must be updated. The lists are fixed for most of them but the “Author” and “Origin” menus that are dynamic. It therefore requires an update function that can deliver a sorted list taking into account each different “Author” or “Origin” in the database.

The solution involves the use of a particular Excel object: the scripting dictionary. A dictionary is a special type of table: to each item is associated a unique key that points to this item. The interest is that when using the same value for the item and the key, it ensures the suppression of all duplicates and empty values. As the “Authors” that can be found in the database will be very redundant and the total number of entries very large, the use of a dictionary presents a great advantage compared to a simple array, especially as the aim of this function is to provide a sorted list for the dropdown menus. However, a dictionary is not adapted to be manipulated like an array. Another transfer is necessary to a temporary array, which can then be sorted. This is easily done with the `List` property of the dictionary (See code below). The sorting is done by a simple algorithm comparing two adjacent values of the array. Once sorted, the dictionary is emptied and updated with the sorted array. As it was declared as a `Public` variable, the dictionary’s list is accessible from all the other modules to display the updated dropdown menus. This is a dynamic solution as the list is calculated once again each time another author is added or one is modified but the data is protected since the dropdown-menus are locked for the user. The following code gives the solution adopted for the “Author” list, the exact same lines being applied for “Origin”.

```vba
...```
Another thing that can be pointed out on the “New_Item” form is the management of the data format. Most of the fields are simple text, the “ATA” is also stored as text to keep the zeros but the “Date” should be a date. The column’s width is already set right for the dropdown menus but the display of fields like “Title” may need to be adapted. “Question” and “Answer” are not displayed but the text needs to be wrapped up in the cell so that a query remains displayed on a single line. Even if these considerations are only small display features, they are an example of the number of things that the programmer has to think of when programming the complete application.

2.2 SEARCH

2.2.1 Search by date

Searching by date is an interesting feature of the search engine. The main issue with filtering by dates is the large number of particular cases one can encounter: non-numeric characters, day’s number above 31, month’s number above 12, no respect of the date format, etc. The following code section presents a way to filter the database by date with the essential use of the DateSerial function and the setting of default values to avoid impossible cases. One can also notice the use of the automatic AutoFilter function with two criteria. This function cannot take more than two different criteria and acts only on one field.
2.2.2 Filtering by keywords

One of the main functions of the tool is to be able to search among the database for any query by filtering by a list of keywords, which is the most common way. As the AutoFilter function has been seen to be limited to one field and two criteria, this is generally done in Excel by use of the embedded function Advanced Filter. However, this function has proven to be very limited in terms of number of keywords. The purpose of the Advanced Filter is to automatically filter a range by a set of criteria indicated precisely in an adjacent table. The criteria can be words, dates, formulas and so on; and depending whether the criteria come in columns or aligned, the logical operators AND or OR are used. This method is therefore not adapted for an unknown number of keywords or if you want to combine the logical operators. The problem was therefore to find a way to program a filter with an unlimited number of keywords.

After several attempts, an unorthodox solution was found. The first step is to separate all the keywords by going through the textbox and detect the spaces. Then, we can use the embedded Find function of Excel to list all the queries that contains anyone of the keywords and set a new attribute of the query to the value “Found” (using therefore another field/column in the table). A simple filter can finally be applied to the database on this last field with the criteria “Found”. Overall this method will give the same results than an OR logical search by an unlimited number of keywords and is much more time/code efficient than using any embedded advanced filter or create a new complex filter. The following code illustrates this method.
2.2.3 Optimization of the search

As the search engine allows many different criteria (several dropdown menus, a search by date and by keywords), some optimization of the SEARCH has been done. As searching in an Excel table is the equivalent of filtering the database to display only the relevant lines, the idea is to use the `Specialcells(xlCellTypeVisible)` special kind of range corresponding to the remaining cells displayed of the database after any filtering action. Instead of filtering the database by each of the criteria, a better solution is to filter only the visible database. The impact will be even greater if many criteria are chosen as the keywords, the heaviest filter, will come last and the overall lapsed time can be divided by a factor of 10 to 1000 depending on the size of the table. However, a problem arises when using this method of refining the database after each criterion. Filters only apply on a continuous database and some of the fields (“Question”, “Answer”…) are not initially displayed. Therefore, to be able to use the `Specialcells` function, display modifications must be done within the module, even if the user will in any case not see any of them.

```
Filtering by Keywords

List_Keywords = Split(TextBox_Keywords.Value)  "Separation of the words and storing them in a vector"  IF Not UBound(List_Keywords) = -1 Then  "If there is at least one keyword"
    With Union(Columns(3), Columns(10), Columns(13), Columns(14))  "Definition of the fields to go through "Title", "Equipment", "Question" and "Answer"
        For i = LBound(List_Keywords) To UBound(List_Keywords)
            Set Cell = .Find(What:=List_Keywords(i), LookIn:=xlValues, _  "Find the keywords one by one in every cell"
                LookAt:=xlPart, SearchOrder:=xlByRows)
            If Not Cell Is Nothing Then  "If the keywords has a match"
                First_Address = Cell.Address
                Do  "On Error Resume Next"
                    List.Cells(Cell.Row, 18).Value = "Found"  "Then set a new attribute "Found" to the query"
                    Set Cell = FindNext(Cell)  "Set Cell = FindNext(Cell)"
                    Loop While Not Cell Is Nothing And Cell.Address <> First_Address
                End If
                Next i
            End With
        End For
    End With
    .AutoFilter Field:=16, Criteria:="Found", VisibleDropDown:=False  "Filter the database with the criteria "Found""
    End If
    Columns(18).Value = ""  "Restoring all the found attributes to nothing"
```

2.3 DETAILS

2.3.1 Several instances of Details

Another advanced feature of the tool that has been required is the possibility to open several Details at the same time, in order to compare them for example. This implies first to have modeless Userforms, as opposed to the modal forms that were introduced previously, meaning that the user can still work on the application while the form is displayed and therefore also open another form. The second step is to create each time a new instance of the form Details instead of using a unique form. The VBA code below highlights the conceptual difference in terms of object programming between an object and its instance.
Then arises a problem linked to the New_Origin and New_Author forms. Although these forms are unique, they now need to be opened from a particular instance of Details and fill the right form when closing. VBA, as opposed to VB, does unfortunately not expose the Handle property that identifies a particular form. A workaround would be to retrieve the Handle of the window through the Windows API functions. However, we saw that for maintenance purpose, it is safer not to use non-Excel functions such as the API functions.

The following code explains the final solution implemented on Tanga_DB to display several forms. Rather than using an inaccessible Handle, this solution passes the whole Userform as a parameter to a subroutine. Even though it uses more cache memory, it enables to perform any action on this particular instance of Details in any other module of the application. We can point out that the subroutine entitled Which_Details has to be called before showing the New_Author form to be able to use the variable “Me”, which would still refer to the Details form and not the New_Author form.

```
Private Sub CommandButton_Add_Author_Click()
    Load New_Author_Details
    Call New_Author_Details.Which_Details(Mo)
    New_Author_Details.Show
End Sub
```

2.3.2 Layout in a Textbox

The possibilities to keep the layout for example when copy/pasting from another source into a textbox or change it directly inside the textbox was investigated to improve the display of long paragraphs in the database, especially regarding the fields “Question” and “Answer”.

- RichTextbox control:
It possesses the same features as a basic textbox but also enables the layout of the text (colour, bold, italic, underlined…). This can be done by showing a short toolbar when the focus is on the RichTextbox. It was already a peculiar control available under a specific library in Excel but Microsoft has now disabled its use.
in Office applications posterior to 2003 because it appeared to be linked with a safety breach. One potential solution to this problem would be to use VB6 to wrap the rich textbox in a container control and use this secure control on the forms. This solution unfortunately requires having VB6 and the specific container control installed on every PC station that would be using the complete program.

- **InkEdit control:**
  Another workaround investigated was the possibility to use an InkEdit control. Being derived from a RichTextBox and still allowed in Office applications, the primary purpose of this control is related to handwriting, through a tactile screen or by direct use of the mouse. The box detects the handwritten input and tries to convert it into the right text while identifying the basic layout (bold, italic and underlined). However, the programmer can only access either the plain text or the code representing the formatted text so that he cannot display directly the formatted text elsewhere than the InkEdit control. This workaround was considered too heavy for the application considering the reduced improvements it would have provided in layout.

![InkEdit control](image)

No satisfying solution was found that enabled to keep the layouts but it was considered as a minor requirement and dropped. The textbox still keeps the carriage returns, the bullet points and the case; and the text when printed is therefore fairly readable.

### 2.3.3 Printing a Userform

Considering the display issues discussed previously for the “Details” of an Item, it was suggested to offer the possibility to the user to print these “Details”, primarily to ease the reading. In VBA, the function `Userform.Print` exists but only prints what is on display on the screen. If a textbox is smaller than the texts it holds, only part of it will then be printed. The solution chosen is to fill a Word template from Excel, placing text at different bookmarks and print it directly on a default printer or as a PDF saved on the computer. It allows the user to keep an easy-to-read version of the queries he is interested in without having to do anything but click on a button. See in Appendix 5 an example of a printed query.

The following code details the steps of using a word sheet via Excel. The first step is create a new Object referring to a word Application in which to open the template. Then, several ways to identify where to copy the text can be used in Word: shapes (WordArt titles for example), tables, bookmarks (which are very useful as they refer to any position in the plain page), etc. The `Printout` function automatically uses the default printing settings. An important step of the code is to reset the variables corresponding to the application and the Word document to release memory space and not open several Word applications.
Like all programming codes, VBA possesses its own features and solutions to optimize the application. In this part are given several pieces of advice that were used to program Excel in an efficient and intelligent way. A large part of these recommendations are related to the typical object structure of Excel that was presented in IV.B.2.

It was seen that the most important class in Excel is the class Range as it refers to a definite number of cells and Excel’s main function is to manipulate these Ranges and change their properties. A first way to optimize the code is to use arrays instead of ranges when heavy manipulations are required on the values: sorting cells by values for example. The programmer declares a dimensional array as a new variable and copies the range into it. He then applies all the necessary changes directly to the array, which are easier to handle and writes the changed values in the range. This solution can be adapted when other properties of the range are concerned such as the background colour or the borders. In parallel, it is important to always reduce the used range to the minimum: using for example functions such as SpecialCells (See 2.2) or by removing all empty values.

Another issue is the class hierarchy of Excel, as many objects have a long list of parents. For example, the property Value is related to a Range, which belongs to a Worksheet, which belongs to a Workbook. In order to avoid going through long paths in the code when calling an object, two different solutions are available:

- **The With function**: The With function is interesting whenever the same root must be used several times within a module. Within the With block, the root designated by the function will be implied and will not need to be written. It enables better reading of the code, but also saving time when running the module, as the long path is kept in memory.
Setting a variable: When an intensive use of a specific object is required, another solution is to create a new variable referring to this object. Using the Private or Public properties, these variables can be visible at a module or application level and are stored temporarily.

An important aspect of using variables is the way of declaring them. The use of the Explicit function at the top of a module imposes to declare variables of a specific type so that Excel avoids having to determine the data type each time the variable is called. Another advantage of this function is for debugging purposes. As each variable is of its own specific type, conflicts or unadvised use of these variables are more easily identified. Particular types of variable are the arrays or tables as their dimensions can be unknown prior to the declaration. The use of the function Redim Preserve enables to resize a table without erasing the values already stored in it.

There are several tricks to reduce the number of instructions in the code. We can give here two of them that have a real impact in terms of computing time:

- The For Each loop: Instead of using a For To loop that uses an index, the For Each loop goes through all the elements of for example a table or a Range. The code is then clearer to read and more efficient.

- Selecting and activating: When using for example, the macro’s recorder from Excel, a programming beginner can find extensive use of the Activate and Select functions that put the focus of the application on a particular object. These functions must be avoided as much as possible in the code as a selection is not required to modify the properties of an object or access its functions. Therefore, they only add more instruction lines without a real contribution to the application.

A large number of controls on a worksheet (Buttons, checkboxes, hypertext links, etc) can be heavy for the application and considerably slow it down. As for this database, many queries can possess between one and three links each, the impact could be significant on the quickness requirement of Tanga_DB. A way around was to store only the addresses in the cells without creating the link and make the link accessible only when opening the DETAILS form of a query.

As Tanga_DB is an Excel based application, most of the objects and properties are related to something visual (see IV.B.2). Running a macro or a few lines of code usually involves many display updating. It is therefore important to turn off everything but the essentials when the code is running such as the updating of the screen (Application.ScreenUpdating Boolean property), the display of the status bar (Application.DisplayStatusBar Boolean property) and the calculation of the worksheet. In a SEARCH for example, there is no need to display all the different steps of filtering but only the final result; and it is also more convenient for the user as all these changes are usually too quick for the eye. It is however important not to forget to reset all these properties at the end of the module. A special care must be taken with the function Application.Calculation. By default, Excel is set to recalculate automatically the cells that have been changed or may be impacted by the change every time the hereby changes occur. Disabling the calculation function can therefore impact the reliability of the application (formula values for example). In the case of Tanga_DB, no cell contains a formula so the calculation can be disabled with a limited risk.

The most time limiting functionalities of the application were presented earlier and are the management of the dynamic “Author” and “Origin” lists (with mainly the sorting of the entries) and the SEARCH by “Keywords” as they involve large loops. Even if optimization methods and the best VBA programming practices have been used in the code to improve the performance, there still might be a significant impact on time delay when the database will be filled, which cannot be assessed properly numerically. Filling the database is besides the next step of the project after the application has been validated.
D. Filling the database

1. Final debugging

If we go back to the V-cycle presented in II.E, we can remember the different steps of testing that were introduced:

- **Unit tests**: This phase was performed with Excel's embedded debugger (detailed step by step, breaking points, etc) until the application could compile the code and run functionally.

- **Integration tests and systems tests**: The review of the prototypes by some users validated the structure and small functionalities of the tool.

- **Acceptance tests**: This is the last phase, corresponding to the commissioning of the tool. The board has validated the final version of the tool, which the users have started to use. A few small mistakes were found, such as an improper “filtering by date” function, but were solved rapidly, proving that the V-cycle has been well respected and large structural changes avoided.

So far, PSI engineers have well adopted this new application that seems to be as close as possible to their needs. A maintenance guide has been written and a VBA teaching session was done with my tutor so that the domain will be able to update the application and solve the main potential sources of bugs in the future.

2. Trace past experience

As was seen in II.E, an additional step is to fill the database with as much past experience as possible. Several questions were raised regarding the methodology and the means to perform this task. We try to give in this part some answers that were decided with the board when commissioning the tool. One must however remember that other possibilities might be considered in the future to fill the database as the following ideas were only suggestions that Airbus might not be able to implement. As of today, the application is fully functional and filled step by step, even though the search function will be limited until a proper solution to trace this past experience is found (See V).

- **The right knowledge**

They are several sources of knowledge that were identified as relevant and accessible to be stored in Tanga_DB:

- Generic topics in the Handbooks: the information contained in the Handbooks is usually very helpful and PSI engineers often refer to them. Entering in the database the different topics covered by the Handbooks will make the reference appear when performing a related search and will direct quickly the user to useful information.

- By the same reasoning, major topics that led to technical notes or fully written reports are worth being entered in the database. Most of this documentation is spread in the servers.

- Start by transferring the queries listed in the limited excel tables that were in use in each section (See III.E.2) and try to complete them with the missing information.

- Associated to this knowledge are the requests the person in position might have treated and remember since they started the job.

- **A limit to go into the past**

- It will be very hard, not to say impossible to trace the experience of persons that used to be PSI engineers. Even if they left any document or e-mails stored before leaving, it will take some time and will be hard to go through them to capitalize the valuable information.
As for the documentation (technical notes, Handbooks, etc), there is no real limit as every relevant documents that can be found should be capitalized.

- **Who to fill the database?**
The knowledge identified above will require however some resources, time and persons, to be entered in Tanga_DB. Here are the solutions suggested:

  o Every PSI engineers must find the time to fill their own treated queries, as much as possible, as they are best placed if not the only one to access quickly the proper information.
  
  o After budget consideration, it has been envisaged to hire a temp via the EPDX domain to fill the database with the technical documentation’s topics. It might require some time to dig and find all the valuable dormant data available in the domain.

With a fully developed tool, it is now time to give an example of its use, assess some of its performance and confirm that the global objectives stated in I.B have been fulfilled.
V. Results

A. Simulation of the new process

An example of the new process detailed in II.B, which is associated with Tanga_DB is given here to better illustrate how the application works. This example is taken from real issues that occurred during my project and even though the database was fairly empty, the engineer found valuable information by using the tool.

1. Receipt of the query

The first step of the process is the receipt of a new query. As they have to be solved within a short timeframe, they always have priority over other tasks the engineer might be conducting at this time. These other actions are dropped and all the focus is put on answering the request. We take here the example of the production query coming from the FAL in Toulouse on transient vibrations that was described in III.A.2. After having detected on the cockpit data a transient peak in vibration on one engine during a pre-delivery flight test, production engineers asked several questions:
- Could we explain the origin of this peak?
- Could we predict the benefit of a fan trim balance and what would be the next step if it failed to improve the vibration state?
- What would be acceptable transient vibration levels/durations?

2. Searches through Tanga_DB

As the database is rather empty, a simple search with the criteria “vibrations” gives only 3 results. Looking through the details of each result, we realize the second query entitled “MSN 5213 – Vibration in climb” deals with the same type of issue: transient peaks out of bounds that could not only be detected through the cockpit controls. This query happened in production as well but for one of the engine of the Long-range family and presents the following fields:

![Figure V.1, Results of the search](image)

- **Question:**
  “During a pre-delivery flight, the engines were snagged because of high transient N1 vibration levels seen during take-off and beginning of climb (high power phases). This phenomenon was observed on the ECAM and reported by the crew, but could not be seen on the EVMU in-flight unbalance report as only stabilized vibrations during cruise phase are recorded. The question asked was: Which actions should be taken considering these vibrations level?”

- **Answer:**
  “Considering significant dissymmetry between both engines during take-off and climb and the margin between vibration levels and the recommended AMM thresholds, numerous fan and LPT rebalances
would not be efficient and could even degrade the situation as no accurate data is available (high transient peaks reporter by the crew different from the acceptable stabilized data found in the reports). It was therefore decided to perform a DMU programming aiming to isolate the vibration issue during take-off and climb and define relevant limits for these flight phases See vibration handbook.”

3. Fill the database

The next step is to solve the query. The mentioned handbook explains the limitations of a fan trim balance in such cases. We saw in III.A.2 that the issue was in fact a faulty connector but the similar query found in Tanga_DB helped answer to why one should do only one fan trim balance and why the transient peaks should be discarded.

Once the issue is solved, the engineers have to enter the query in the database with its solution. Figure V.2 shows the users filling the query.

![Figure V.2, New query in Tanga_DB](image)

In general, there can be different situations possible regarding the use of Tanga_DB:

- A similar query has already been treated in the past providing sufficient information for the treatment of the new question. The person in charge of the previous query might still be in position to provide direct help.
- Queries can be found to be related to the topic (same system for example) but differ too much to be completely useful. One could however find interesting references such as a Handbook for example.
- Nothing can be found on the topic and the engineer must find another source of information before the commissioning of the application, the engineers were facing the third scenario all the time. They now have a possibility to know quickly if they can find additional help in the database.

B. Performance

It is interesting at this point to state again the different objectives of the application and read them with all that was detailed about the tool in mind.

- Improve time response, accuracy and exhaustiveness of daily queries treatment
- Tracing the experience for Airbus
- Increase the autonomy
- Support to new comers
- Less stressful environment
- Share the experience among programs

The first month of using the database has proven the following points related to performance:

- **Filling the database:**
  o No impact the work of the persons: very quick to fill
  o Tends to present a more organized final answer
  o Homogenization of the work among the whole domain

- **Searching:**
  o Available information quickly for the users
  o The efficiency of the tool increases with the number of entries available in the base
  o As was seen in IV.D.2, past experienced is not yet included, also limiting the performance of the search engine

Every query that is entered in the database is also a gain of knowledge capitalized for Airbus. It might serve not only supporting current programs but know what kind of problems to expect for the new programs: the A320neo and the A350 XWB. EPDX participated in the development of this database and will monitor its use as it goes in parallel with their activities.

Finally, another interest of the tool was detected through its development. It serves indirectly a budget purpose. The board needs to estimate each year the amount of activity done by the domain in order to challenge the next year’s budget allocations. Tanga_DB allows counting the number of problems solved in several categories (Customer service and Continuous support), provided the tool is rigorously filled, and provides more accurate business drivers to the hierarchy.
Conclusion

With increasing market constrains and competition, knowledge management is nowadays a key issue for companies to ensure their success. Despite this importance, even greater for larger companies, there is a huge step between acknowledging basic knowledge management principles and implementing long-term practical procedures. In Airbus’s propulsion systems integration domain, engineers agreed that they would benefit from capitalizing a specific type of activities: in-service and production queries. Hence, an application was developed to meet the specifications established by the PSI engineers as potential users. The focus was made on getting the persons involved in the project. The tool has already been used for a month by the domain and reached the level of performance that was expected: improve the time to respond to a request, support new comers and experienced engineers through a new source of information and not impact the work of the engineers thanks to its quickness and simplicity of use.

A remaining unknown is now whether this tool will continue to be used in the long-run. Will the PSI engineers appreciate this additional help or drop the database for some reason: the need evolves for example. Through this first month of testing and using the database with my support, they have become acquainted with the tool and many realized its advantages. In addition, a lot of pressure came from the hierarchy to make sure that the engineers comply with this new process. It was for example included in each group leader’s objectives. However, ensuring the continued existence of Tanga_DB requires a personal effort and rigour from every user. If they do not want to force themselves to use it, they will not use it. Knowledge management is an important issue for every company but is often limited by the fact that the employees must be at the origin of the capitalization, which is rarely their primary task.
References


Available at “http://www.unc.edu/~sunnyliu/inls258/Introduction_to_Knowledge_Management.html”

[5] Cultivating and formalizing a culture of knowledge capitalization, Oracle, 2010


Available in the Intranet at “http://km”


[16] Starter motor events - Cluster gear fractures, IAE, November 2012

[17] Starter improvements, IAE, October 2012


[20] SEE Requirements for engineering support, Airbus Service Level Agreement, 2008


Available at “http://msdn.microsoft.com/en-gb/office/fp123562”


[27] *IAE V2500 Accessories guide*, International Aero Engines, 2011
Appendices

Appendix 1: IAEV2500 Pneumatic starter

Here is a short description of the pneumatic starter taken from the IAE V2500 accessories guide [27]. This technical document is not official but may give additional information to the approved publications, regarding for example the manufacturer of the part, a 3D drawings, the part numbers, the SB associated to the part and so on and so forth.
Appendix 2: Tanga_DB – User Guide

Activate the Macros before any other action
- Save the spreadsheet before closing or, in case of unwanted manipulation or program error, close without saving.

Protect the database by using only the buttons and following functionalities

Fill in a new query in the database
Quickly fill part of or a complete new Query in the database. To ensure some homogeneity of the data, the fields marked with (x) are mandatory. The other fields are here to provide more information, useful for a future search. The Title is divided into three parts: MSN xxxx & Airline XXX – Object – Defect.

Tips:
- Origin and Author: If your entry is not already in the dropdown menus, click on the + to create a new one. Please respect the format for Author.
- Date: Respect the format: dd/mm/yyyy
- Copy/Paste is possible in all textboxes but the layout (bold, italic, underlined, colour...) is lost.
- Question / Answer textboxes:
  - No picture or table can be included in your text. If they are valuable information, please consider a Link.
  - The character limit is 32767. If your text is longer, please consider a Link.
- Hypertext Links:
  - Create a file in the archive folder associated with the database or reference a document in the dedicated space on the server. As Excel does not support all formats, referring to a folder is preferred.

Look for solved queries matching your criteria
Filter the database with different criteria and find a specific query or similar covered items matching your search. The results will be sorted by program and most recent queries. The last Search is saved, use Clear to reset.

Tips:
- Keywords:
  - Write an unlimited number of Keywords separated by spaces. Case is not taken into account. One should however ban any punctuation (/ or ; for example).
  - The search will find any result containing at least one of the Keywords.

Display the complete database

Consult: Double-click on a Query to Consult. Several Queries can be opened at the same time to compare their content.
- Modify: Click on any Textbox to modify or complete your Query.
- Links: Use the arrows to follow the hypertext Links if any.
- Print: Print automatically the selected Query as a PDF available in the application's folder.
- Delete: Deleting a Query is irreversible.

Sort: Double-click on a Header to sort the database by this header.

Category: Double-click on the category cells to access the complete division of the activities.
### Appendix 3: ATA breakdown

<table>
<thead>
<tr>
<th>ATA Breakdown</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Time Limits - Maintenance Checks</td>
</tr>
<tr>
<td>6</td>
<td>Dimensions and Areas</td>
</tr>
<tr>
<td>7</td>
<td>Lifting and Shoring</td>
</tr>
<tr>
<td>8</td>
<td>Leveling and Weighing</td>
</tr>
<tr>
<td>9</td>
<td>Towing and Taxing</td>
</tr>
<tr>
<td>10</td>
<td>Parking, Mooring, Storage and Return to Service</td>
</tr>
<tr>
<td>11</td>
<td>Placards and Markings</td>
</tr>
<tr>
<td>12</td>
<td>Servicing</td>
</tr>
<tr>
<td>20</td>
<td>Standard Practices - Airframe</td>
</tr>
<tr>
<td>21</td>
<td>Air Conditioning</td>
</tr>
<tr>
<td>22</td>
<td>Auto Flight</td>
</tr>
<tr>
<td>23</td>
<td>Communications</td>
</tr>
<tr>
<td>24</td>
<td>Electrical Power</td>
</tr>
<tr>
<td>25</td>
<td>Equipment/Furnishing</td>
</tr>
<tr>
<td>26</td>
<td>Fire Protection</td>
</tr>
<tr>
<td>27</td>
<td>Flight Controls</td>
</tr>
<tr>
<td>28</td>
<td>Fuel</td>
</tr>
<tr>
<td>29</td>
<td>Hydraulic Power</td>
</tr>
<tr>
<td>30</td>
<td>Ice and Rain Protection</td>
</tr>
<tr>
<td>31</td>
<td>Indicating/Recording Systems</td>
</tr>
<tr>
<td>32</td>
<td>Landing Gear</td>
</tr>
<tr>
<td>33</td>
<td>Lights</td>
</tr>
<tr>
<td>34</td>
<td>Navigation</td>
</tr>
<tr>
<td>35</td>
<td>Oxygen</td>
</tr>
<tr>
<td>36</td>
<td>Pneumatic</td>
</tr>
<tr>
<td>38</td>
<td>Water/waste</td>
</tr>
<tr>
<td>46</td>
<td>Information Systems</td>
</tr>
<tr>
<td>49</td>
<td>Airborne Auxiliary Power</td>
</tr>
<tr>
<td>51</td>
<td>Standard Practices - Airframe</td>
</tr>
<tr>
<td>52</td>
<td>Doors</td>
</tr>
<tr>
<td>53</td>
<td>Fuselage</td>
</tr>
<tr>
<td>54</td>
<td>Nacelles/Pylons</td>
</tr>
<tr>
<td>55</td>
<td>Stabilizers</td>
</tr>
<tr>
<td>56</td>
<td>Windows</td>
</tr>
<tr>
<td>57</td>
<td>Wings</td>
</tr>
<tr>
<td>70</td>
<td>Standard Practices - Engine</td>
</tr>
<tr>
<td>71</td>
<td>Power Plant</td>
</tr>
<tr>
<td>72</td>
<td>Engine</td>
</tr>
<tr>
<td>73</td>
<td>Engine Fuel and Control</td>
</tr>
<tr>
<td>74</td>
<td>Ignition</td>
</tr>
<tr>
<td>75</td>
<td>Air</td>
</tr>
<tr>
<td>76</td>
<td>Engine Controls</td>
</tr>
<tr>
<td>77</td>
<td>Engine Indicating</td>
</tr>
<tr>
<td>78</td>
<td>Exhaust</td>
</tr>
<tr>
<td>79</td>
<td>Oil</td>
</tr>
<tr>
<td>80</td>
<td>Starting</td>
</tr>
</tbody>
</table>
Appendix 4: Visuals of the "List" page
Appendix 5: Example of a printed “Details”

**BACKGROUND:**
CHH raised a question about the engine vibration on CFM and on IAE on CHH A320 fleet.

**QUESTION:**
1. Is it possible that the resonance phenomenon will be triggered between the A/C frame and the engine due to engine abnormal vibration?
2. If the answer is yes, could you please provide us with the possible range of the engine vibration value?

**ADDITIONAL INFORMATION:**
By researching in our database, we have found that there was a study performed a few years back due to issues in flight line on CFM56-5B- various measurements were taken on aircraft and link to cavity mode response of aircraft to engine. Ref TDO 17013/2000:

****QUOTE****
44 Hz sensitivity is inherent to the single aisle aircraft geometry. It is not linked to a design or quality problem of engine or aircraft when each one is looked apart, but is related to engine/aircraft coupling.
A mechanical excitation around 44Hz, transmitted through the A/C structure, excites the fuselage cavity mode(s):
Radial cavity first mode response of the AIRBUS Single Aisle aircraft.
****UNQUOTE****

Could you please confirm that the results of the above-mentioned study are applicable to the request from CHH for the CFM56-5B engines?
And what about the IAE engines

**Answer:**
A specific study was conducted in 2000/2001 after low frequency noise and vibrations in cockpit and cabin were experienced on newly delivered aircraft equipped with CM56-5B engines. RFW D5235/01 covered this activity. Analysis and tests pointed out that the following root causes:
There is a mechanical excitation around 44 Hz, transmitted through the A/C structure that excites the fuselage cavity modes. This A/C sensitivity is inherent to its geometry (fuselage diameter), and,

- 44 Hz, corresponding to 51% N1, appears to be a characteristic vibration frequency of the CFM56-5B engine. This has been evidenced by recording the N1 vibration levels and the corresponding frequency on approximately 166 engines. Results indicate that a non negligible number of engines had their max N1 vibration levels recorded at 44 Hz.

However, this vibration levels are considered as “normal” vibration levels and are all far below advisory. In order to improve the situation in FAL and in-service a new fan trim balance procedure at 51% N1 has been issued at that time.

This problem was not reported on aircraft equipped with IAE engine. Thus, no study was conducted and we have no evidence that this phenomenon cannot be encountered on this engine.

That’s currently the only known possible engine-aircraft “coupling” or resonance effect which might be experienced with CFM56-5B engines.

Note: there is actually another low vibration frequency case at 27Hz (detectable in the cabin), but it is linked to an old HMU standard P11 which should not be in service any longer.

No study has been performed in case of abnormal vibration levels, which might be consecutive to an engine failure or abnormal operation. A case-by-case analysis would be required should we be able to identify/define a specific abnormal condition.