GEOMETRIC FEATURE EXTRACTION IN SUPPORT OF THE SINGLE DIGITAL THREAD APPROACH TO DETAILED DESIGN

A Thesis
Presented to
The Academic Faculty

By

Aroua Gharbi

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Computational Science and Engineering and
Aerospace Engineering in the
School of Computing

Georgia Institute of Technology

December 2016

GEOMETRIC FEATURE EXTRACTION IN SUPPORT OF THE SINGLE DIGITAL THREAD APPROACH TO DETAILED DESIGN

Approved by:

Professor Dimitri Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Duen Horng Chau
College of Computing
Georgia Institute of Technology

Dr. Simon Briceno
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: November 18, 2016

The only true wisdom is in knowing you know nothing.

Socrates

I dedicate this thesis to my parents for their support, love and trust!

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor Prof.Dimitri Mavris and Dr.Simon Briceno for their insightful comments and useful guidance through the whole project.

Special thanks go to Darshan Sarojini, Evanthia Kallou, Dustin James Harper, Victor Petitgenet and David Rancourt for their contribution to the elaboration of STAnDD framework and sharing their expertise in structures design and optimization.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Aircraft design is a multi-disciplinary and complicated process that takes a long time and requires a large number of trade-offs between customer requirements, various types of constraints and market competition. Particularly detailed design is the phase that takes most of the time due to the high number of iterations between the component design and the structural analysis that need to be run before reaching an optimal design.

In this thesis, an innovative approach for detailed design is suggested. It promotes a collaborative framework in which knowledge from the small scale level of components is shared and transfered to the subsystems and systems level leading to more robust and real-time decisions that speed up the design time. This approach is called the Single Digital Thread Approach to Detailed Design or shortly STAnDD. The implementation of this approach is laid over a bottom-up plan, starting from the component level up to the aircraft level. In the component level and from a detailed design perspective, three major operations need to be executed in order to deploy the Single Digital Thread approach. The first one is the automatic geometric extraction of component features from a solid with no design history, the second phase is building an optimizer around the design and analysis iterations and the third one is the automatic update of the solid. This thesis suggests a methodology to implement the first phase.

Extracting geometric features automatically from a solid with no history(also called dumb solid) is not an easy process especially in aircraft industry where most of the components have very complex shapes. Innovative techniques from Machine Learning were used allowing a consistent and robust extraction of the data.

This thesis will start with an introduction and a motivation for the problem. The second chapter is dedicated to the problem formulation and how the framework created will help tackling some of the problems in the aircraft detailed design. The third chapter gives a detailed explanation of the methodology behind STAnDD before describing the implementation of the automatic feature extraction. The thesis concludes with some results and

suggested future work.

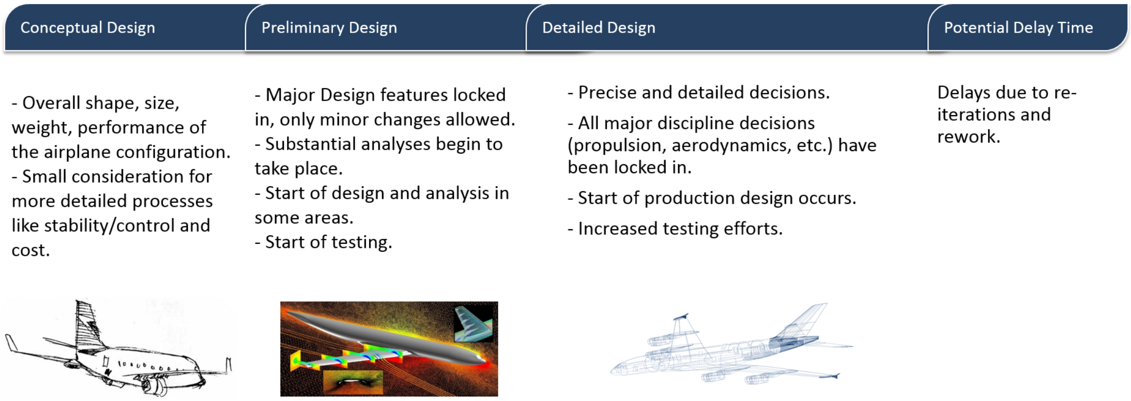
CHAPTER 1
INTRODUCTION AND BACKGROUND

1.1 Overview and Motivation

The commercial aircraft industry is large and growing contributing greatly to the economic growth and spurring multinational collaboration and investment. The framework of such an industry encompasses various domains of which two are the most relevant to this thesis: design processes and engineering processes. The design process of a commercial aircraft is long and complicated taking in average 7 to 15 years. It is usually divided into three main phases: conceptual design, preliminary design and detailed design followed directly by manufacturing which triggers the production process. As shown in Fig.1.1, conceptual design is the phase during which an overall shape, size and weight of the airplane has been estimated. During preliminary design, major features are locked in allowing the start of design, analysis and testing in some areas. In detailed design, precise and specific decisions are made. The production design starts as well as increased testing efforts in preparation for manufacturing. In each one of these phases, reiterations and reworks are often needed in order to reach an optimal design, which leads to an additional delay time that impacts the whole design process.

These delay times are the main reason behind late deliveries. In aircraft industry the overrun costs are as high as 48% as shown in Fig.1.2 with billions of dollars for contractual penalties (Fig.1.3). With an estimate of 20% growth in global demand, some innovative approaches must be included to reduce these costs and delays as much as possible.

There is a wide range of reasons leading to these delays. One of them comes from the fact that the design process is complicated and multidisciplinary. In Fig.1.4, some of these disciplines are given along with the evolution of how much knowledge about each of



Conceptual Design

- Overall shape, size, weight, performance of the airplane configuration.
- Small consideration for more detailed processes like stability/control and cost.

Preliminary Design

- Major Design features locked in, only minor changes allowed.
- Substantial analyses begin to take place.
- Start of design and analysis in some areas.
- Start of testing.

Detailed Design

- Precise and detailed decisions.
- All major discipline decisions (propulsion, aerodynamics, etc.) have been locked in.
- Start of production design occurs.
- Increased testing efforts.

Potential Delay Time

Delays due to re-iterations and rework.

Figure 1.1: Design Cycle of an Aircraft[1][2][3]

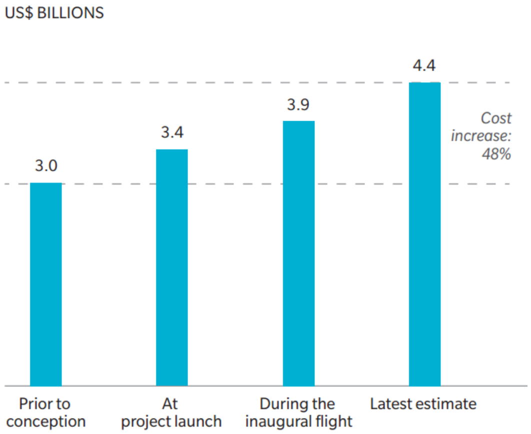


Figure 1.2: Example of Aircraft Development and Penalties[4]

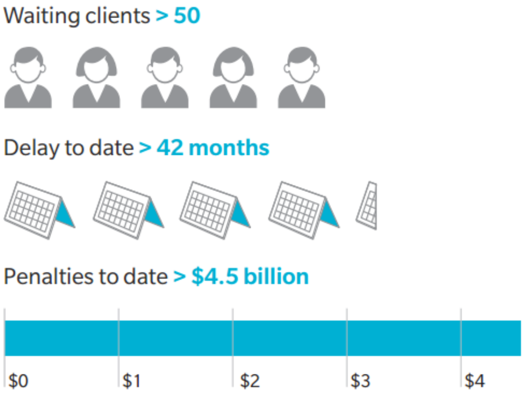


Figure 1.3: Recent Aircraft Program Development Costs, from Preliminary Design to 2014[4]

these disciplines is acquired throughout the whole process. With such uneven distribution of knowledge, uncertainty is prevalent especially at the early stages of the design. This uncertainty implies a large design space that decreases as the process advances. Exploring this design space and eventually reaching the optimal point that satisfies all the constraints is a tedious task calling out a large number of experts. In this space, there are numerous explored and unexplored options that need to be considered. Besides being a very hard optimization problem to solve on its own, the aircraft has a large number of parts that need to be designed separately while keeping in mind the constraints and requirements of the big assembly. There is no tool capable of solving such an enormous design problem holistically. Hence, there has to be some trade-offs and iterations between the parts design and the aircraft assembly design until a convenient solution is reached. Human expertise in this process plays a very important role in assessing these iterations and even reducing their number by making appropriate decisions. In an industry that is worth billions of dollars and in which human safety is a primary concern, there is very little room to make mistakes.

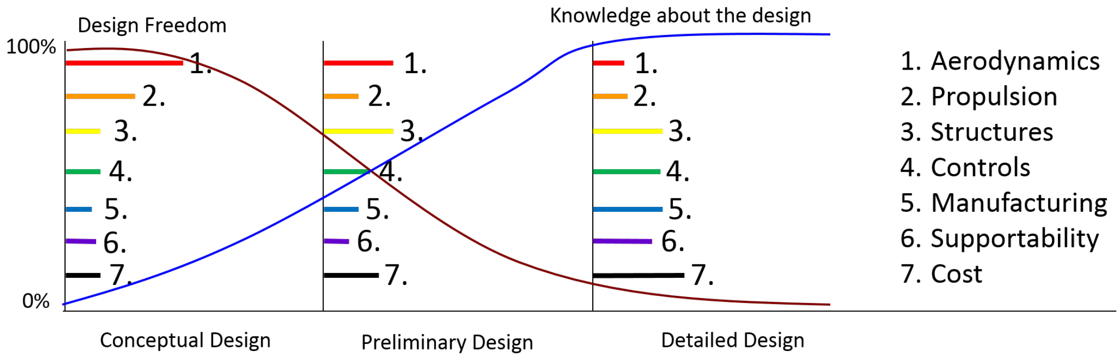


Figure 1.4: Uneven Distribution of Knowledge in the Design Process[5]

The design processes in aircraft industry are guided by the engineering processes which are the operations that need to be conducted in order to solve an engineering problem. There are three main processes that are widely known and used: Traditional Engineering, Concurrent Engineering and Collaborative Engineering. Traditional Engineering is focused on the product functionality. The operations are executed in a ”waterfall” fashion where each operation is conducted separately by a separate team without much influence from all the different operations (Fig.1.5). It is usually referred to as ”over the wall” engineering as

Table 1.1: Comparison Between the Different Engineering Processes [8]

Characteristic	Traditional	Concurrent	Collaborative
Timeframe	Sequential	Overlapped	Shared
Teams	No	A few	Unique
Focus	Product Design	Reduce time	Customer
Objective	Design for Functionality	Design for Assembly	Virtual Manufacturing

well. Concurrent Engineering emphasizes on the parallelization of tasks through an integrated product development process. The teams that work on the different operations are cross-functional working on several tasks at once (Fig.1.6). It simultaneously integrates engineering of all design, manufacturing and operational aspects of a project [6]. Despite its more integrated aspect, ”the wall” mentioned earlier still exists but is not so high. And finally, Collaborative Engineering promotes the idea of one team working on one deliverable. It is defined as the process of having multidisciplinary stakeholders working together to develop a common product[7](Fig.1.7). Table 1.1 highlights the main differences between the engineering processes mentioned above in a commercial aircraft company [8].

From the definitions above, Collaborative Engineering provides the best solutions in terms of reducing the uncertainty at earlier stages of design. Additionally, Collaborative Engineering comes with advanced technologies and a well integrated framework that provides the most complete description of an aircraft. It is a relatively new concept though and for most of the major aircraft companies, it can only be implemented in new projects. Older projects are confined to old mechanisms some of which might be outdated. Interestingly, these old projects are usually the ones that are the most profitable for the companies. For instance, as of August 31, 2016, the Airbus A320 has by far the largest number of orders of all Airbus products (12,776), compared to those for the A380 (319) and A330, A340, and A350 XWB combined (2,820) as shown in Fig.1.8. As for Boeing, the 737 which first took off in 1968, has a total order of 13536 which is more than half of the total order that the company has received since 1955 until September 2016 as in Fig.1.9. This leads to the conclusion that improving the design processes of these aircraft models will result in huge benefits for the company.

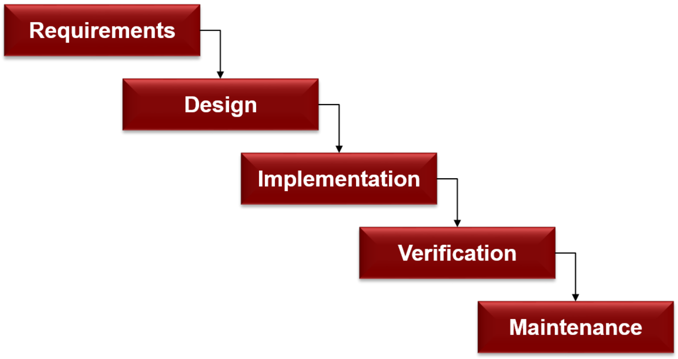


Figure 1.5: Traditional Engineering[9]

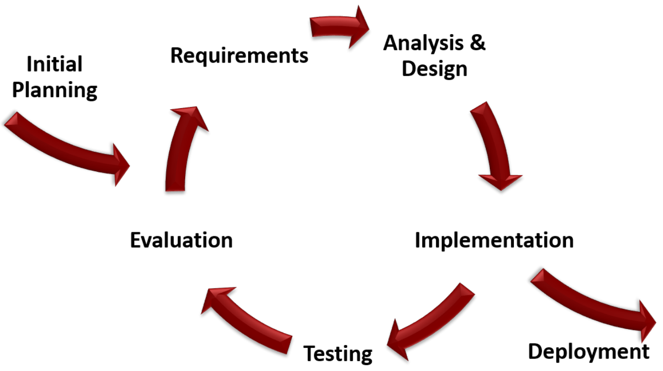


Figure 1.6: Concurrent Engineering[9]

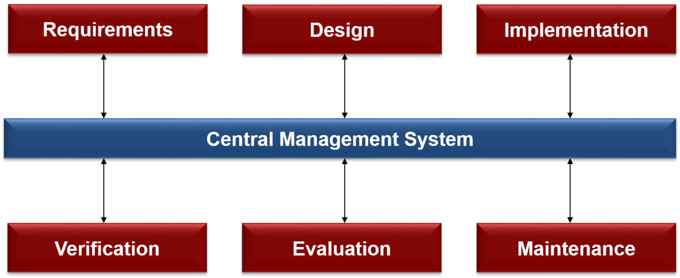


Figure 1.7: Collaborative Engineering

Changes in design process in commercial aircraft industry is not as evident and easy as it might seem. First, these processes are very complex themselves requiring extra care whenever a change is intended as to consider all the variables, risks, constraints etc. The second reason is the large scale of these companies. They usually have multiple sites all over the world. This diversity has unquestionably enormous benefits and provides the companies with a wide range of talents, but when integrating a remarkable change in the processes, this diversity might slow down the integration if not make it impossible. Evidently, each product and aircraft within a company has its unique process. Thus, newer products usually have the advantage of being designed or manufactured using cutting edge technologies and very advanced tools therefore there is not much space for improvement for these products.

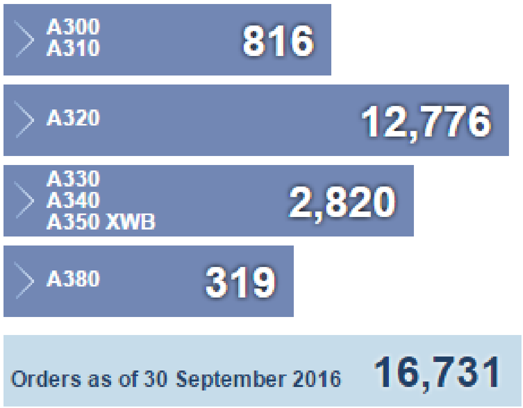


Figure 1.8: Orders of Airbus product lines [10]

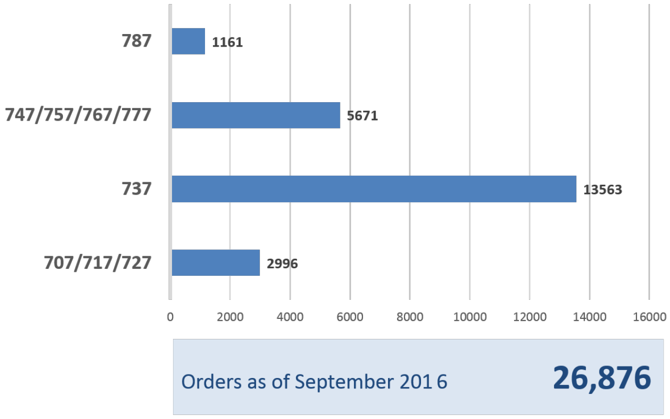


Figure 1.9: Orders of Boeing Product Lines as of September 2016[11]

As much as the idea of introducing new technologies in such complicated process sounds challenging, it is not impossible to execute. An excellent understanding of the existing processes, their advantages and weaknesses is a must before attempting any improvement. Any improvement suggested needs to have a very clear framework, implementation strategy and domain of application as to allow the assessment, the verification and the validation of such changes.

1.2 Research Observations

By examining the design process in Fig.1.1, detailed design is the most demanding and complex in the overall process. This is because it is the phase during which each single part of the airplane is designed and analyzed individually as a preparation to be manufactured. Any improvement in any of these individual and smaller-scale design processes will evidently impact the larger one.

The discussion in the previous section leads to the conclusion that in order to improve the detailed design process, changes must consider the Traditional Engineering methods used as well. Therefore any solution suggested needs to adapt to the context in which it is implemented. The argument was made that Collaborative Engineering offers the best solutions for detailed design but many factors inhibit the full integration of Collaborative Engineering such as the different of tools used for different operations and the involvement of so many disciplines. This doesn't mean that filling in the gaps in Traditional Engineering using concept from Collaborative Engineering is not possible which is the main idea behind developing a method called Single Digital Thread Approach to Detailed Design (referred to as STAnDD). STAnDD leverages both software and process solutions to reduce cycle time in detailed design, support the decision making process and integrate all the disciplines in one single thread. Before further explaining this approach, a literature review needs to be conducted to explain the Digital Thread concept and how it was applied.

1.3 Literature Review

The Digital Thread is a concept that aims to the creation of multi-disciplinary, common digital surrogates for dynamic assessment of the system capabilities [12]. This assessment produces more informed decisions in the different phases of the product development such as preliminary and detailed design, manufacturing, testing etc. Digital Thread comes with the idea of integrating advancing technologies such as Big Data, Cloud Computing and Machine Learning to address some of the challenges that companies and particularly in this case the aircraft industry is facing. A study by LNS research has proven a 46% improvement of forecast across the supply chain in 300 companies only through the use of Big Data [13].

The US Department of Defense has carried various studies to assess the adoption of a Digital Thread approach in their processes. Studies of improvement of Agile Manufacturing technologies using Digital Thread in the US Air Force asserted important benefits such as the quantification of risks at critical decision points and the informed trade space exploration in design and manufacturing [14]. The High Performance Computing Modernization Program Computational Research Engineering Acquisition Tools Environment for Air Vehicles (shortened as HPCMP CREATE^{TM}-AV) was developed as a key enabler of the Air Force Digital Thread [15]. As part of the CREATE-AV project, it is designed to develop and support a set of multi-disciplinary physics-based simulation software products for the engineering workforces supporting air vehicle acquisition programs. The studies carried in this project indicate potential reductions in the overall acquisition time and costs once the Air Force Digital Thread/Digital Twin is fully implemented.

The Digital Thread approach comes with the idea of improving existing design processes. The Aerospace Systems and Design Laboratory in the School of Aerospace Engineering in Georgia Tech has a lot of expertise in improving these and allowing more insightful predictions in earlier phases. One of the relevant methodologies to this project is the generic CE/IPPD (Concurrent Engineering/Integrated Product and Process Development) methodology which was developed to integrate design and manufacturing to reduce cost and produce more efficient designs[16]. The flow of this method given in diagram 1.10 which depicts the different activities in the product lifecycle from conceptual design to manufacturing. The IPPD allows parallel product-process design trades at the system level, the component level and the part level. This is all using techniques and capabilities for both products and processes.

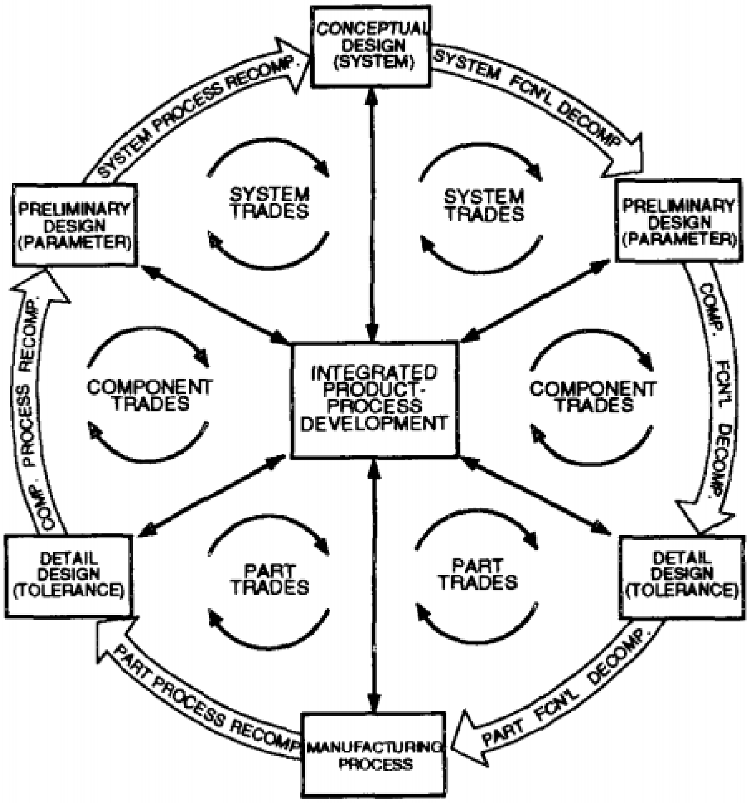


Figure 1.10: Integrated Product and Process Development[16]

The implementation of IPPD can be guided by the methodology in Fig.1.11 which was applied for NASA's HSCT (High Speed Civil Transport) concept. It institutes the benefits from the successful methods and tools for products and processes in a Multidisciplinary Design Optimization framework to generate feasible alternatives taking the economic impact of the aircraft as comparison criterion [16].

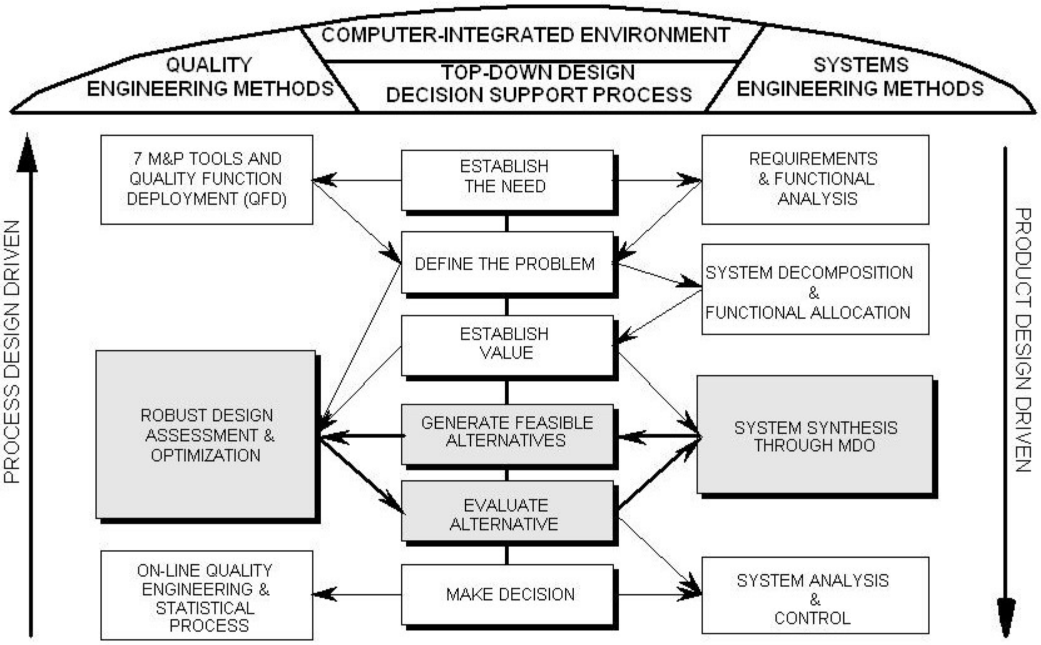

COMPUTER-INTEGRATED ENVIRONMENT
QUALITY ENGINEERING METHODS
TOP-DOWN DESIGN DECISION SUPPORT PROCESS
SYSTEMS ENGINEERING METHODS
PROCESS DESIGN DRIVEN
PRODUCT DESIGN DRIVEN
7 M&P TOOLS AND QUALITY FUNCTION DEPLOYMENT (QFD)
ESTABLISH THE NEED
REQUIREMENTS & FUNCTIONAL ANALYSIS
DEFINE THE PROBLEM
SYSTEM DECOMPOSITION & FUNCTIONAL ALLOCATION
ESTABLISH VALUE
ROBUST DESIGN ASSESSMENT & OPTIMIZATION
GENERATE FEASIBLE ALTERNATIVES
SYSTEM SYNTHESIS THROUGH MDO
EVALUATE ALTERNATIVE
ON-LINE QUALITY ENGINEERING & STATISTICAL PROCESS
MAKE DECISION
SYSTEM ANALYSIS & CONTROL

Figure 1.11: Georgia Tech Generic IPPD Methodology[16]

In terms of technology use, ASDL has also developed various comprehensive methods for decision making in the early stages of commercial aircraft design. The Technology Identification, Evaluation, and Selection (TIES) for example, is a methodology which provides a framework of technically feasible and economically viable alternatives in a nine steps process illustrated in Fig.1.12. Applied on a 150 passenger, intracontinental, medium range transport as a proof of concept, the method suggested advances in propulsion technology to improve the its performance and economic figures of merit.

The Digital Thread approach promotes the use of advanced technologies in simulation, Big Data and Machine Learning. Outside the scope of a Digital Thread, these technologies have already been used to solve various problems in aircraft industry. Big Data is one of the most common solutions in aviation since there is a massive amount of data like flight tracking, safety reports etc around an aircraft. All this data needs to be managed in the most efficient way. As an example, in a twin-engine Boeing 737, over 2×10^9 terabytes of data are generated yearly from the engines' sensors in cross-country flights. This large-scale, unstructured and non-homogeneous amount of data is a perfect candidate

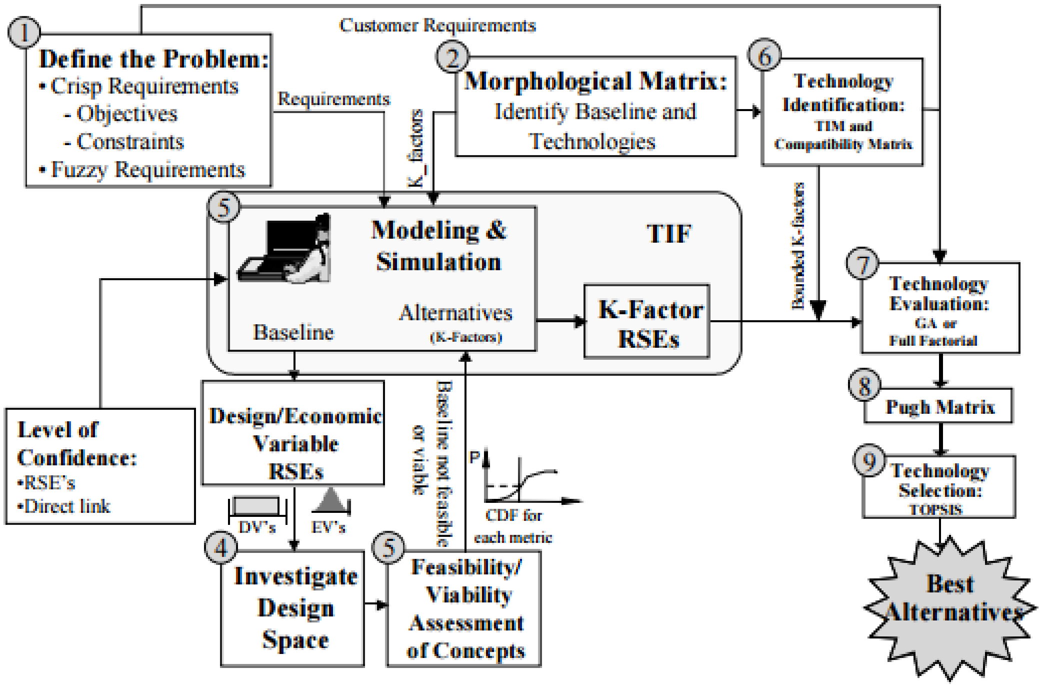


Figure 1.12: Technology Identification, Evaluation and Selection Method[17]

for Big Data solutions. Companies need such data to derive insightful information and make better predictions in real-time systems. Big Data analysis can be used to predict faults in the components which can help improve the Condition/Preventive maintenance procedures [18].

In addition to Big Data, several other innovative approaches in Machine Learning were explored to solve problems in aviation. Supervised classification for example has been widely used. J.Williams has developed a methodology to fuse data from different source to produce a real-time diagnosis of turbulence [19]. The algorithm was trained using a retrospective dataset that includes turbulence reports from commercial aircraft and collocated predictor data. The classification was made using the random forest classifier which showed better results than logistic regressions for the same problem. This classifier helps avoiding turbulence which is a major cause of flight delays that cost airlines millions of dollars every year in compensation.

Another example of supervised classification can be brought up from N.Oza's work in classifying system health and safety documents [20]. A Support Vector Machine with Simulated Annealing and a Non-negative Matrix Factorization were used to classify the Aviation Safety Reporting System and the Aviation Safety Action Plan documents. These methods proved to be highly accurate and relatively stable which is of great importance considering the volume of documents that need to be categorized.

Machine Learning has also been present in air traffic control to predict the trajectories of airplanes as in the work done by M.Hrastovec and F.Solina [21]. Based on a multi-dimensional data base, a nearest neighbor algorithm is run to calculate the average performance of a set of aircraft flights similar to the one under investigation. The similarity is specified using a various number of attributes that are not always available. This predictor has showed a prediction accuracy that surpasses the methods currently used by air traffic controllers.

The use of Machine Learning in the actual design cycle process of the aircraft (Fig.1.1) isn't really popular in commercial aircraft industry and this is for some of reasons mentioned in the previous section. The process of integrating and certifying a new tool or methodology can be long and daunting. Nonetheless, there were some attempts to do so as small projects like the work done by A.Variyar in trying different regression techniques including a Gaussian process regression and a multilayer perceptron network to predict aircraft performance in conceptual design phase [22].

Part of thesis deals with the automatic feature extraction from CAD models. This topic has been already explored by researchers. The extraction techniques varied based on the extension of the solid used. P.Arunkumar and al. used a STEP approach in extracting product features from CAD solids [23]. The step file is analyzed based on "3-Edge Theory" to extract simple and complex features. Bhandarkar and Nagi have also leveraged the use of STEP models as the means for neutral form exchange of product related data between 'partner' companies in the context of Agile Manufacturing [24]. A C++ software was written to extract features from solids that can be manufactured by milling process. The system loops through every component of the model and stores the feature data in a format that can be transmitted between different locations. Bohm and al. have developed a system that uses range images to detect features[25]. The estimates of mean and Gaussian curvatures of each pixel in the range image are obtained from a least squares surface fitting algorithm. Then a simple minimum distance classification is implemented to assign pixels to the surface label of the closest feature. The method can process different types of surfaces and helps establishing a CAD model-based object recognition system for industrial parts.

CHAPTER 2
PROBLEM FORMULATION

2.1 Detailed Design of a Commercial Aircraft

Detailed design is the phase during which the actual design of the parts that will be manufactured takes place [26]. This phase is usually the longest, the most expensive and the largest in terms of human resources needed in the whole design cycle. An overview of the framework of this phase is provided in Fig.2.1.

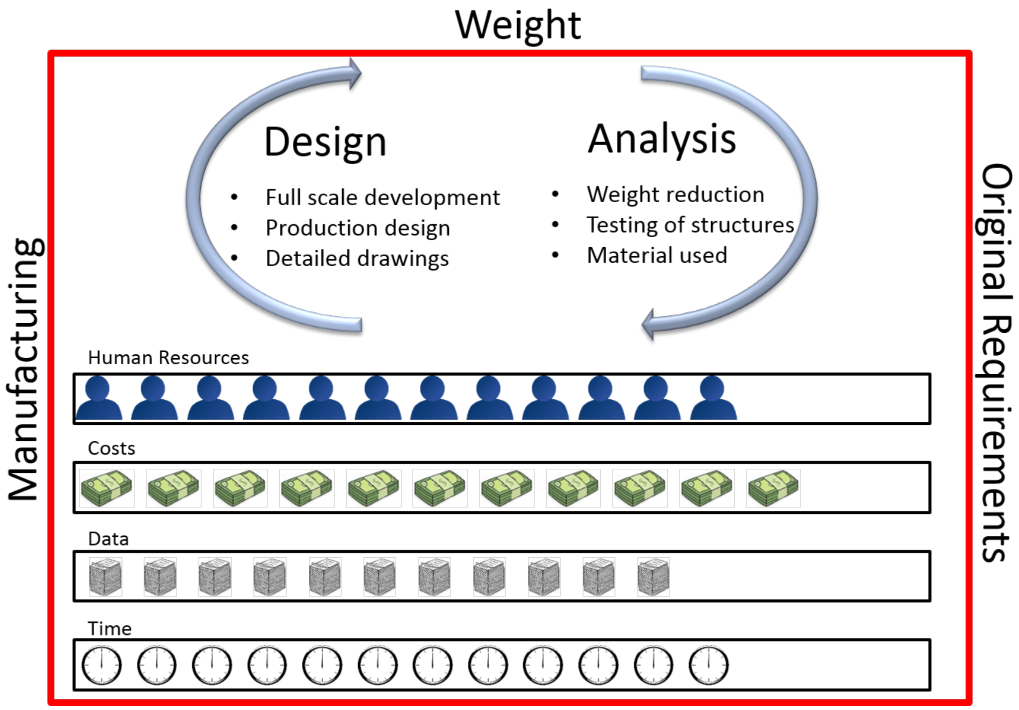


Figure 2.1: Detailed Design Framework

Detailed design is concerned with components, subsystems and systems. Each level has a separate set of constraints, design principles and tools however the final assembly must meet the original requirements that were already set in earlier phases of design. The distinction between these different levels is very important in STAnDD as the solution suggested proposes an implementation strategy based on the scale of each level. The constraints for each level are numerous: loads, manufacturability, cost, weight etc.

In every level, many constraints need to be accounted for: loads, manufacturability, cost, weight etc. Even though the exact operations conducted during this phase are different from one company to the other, they can generally identified within six different disciplines: sizing, analysis, prototyping, testing, validation and manufacturing (Fig.2.2). Among this list of disciplines, two major ones are identified: design and analysis. Design includes the preparation of the actual drawings, the CAD (Computer Aided Design) files and every task that exactly defines the geometries and dimensions of the parts in full details. In the subsystems and systems level, the assembly contains all the details about each component, their locations, parts used to attach the different components and so forth. The structural analysis is concerned with stress calculation and identifying the areas of failure or improvement.

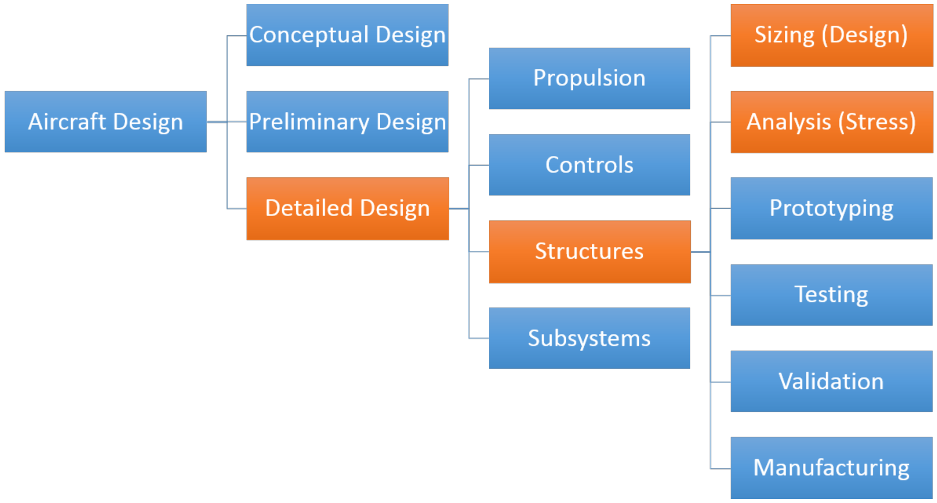


Figure 2.2: Breakdown of Disciplines in Aircraft Design

In most aircraft companies, Computer Aided Design software such as CATIA or Siemens NX are used. These software offer powerful 3D design tools and support different stages of product design mainly engineering and manufacturing. This is useful for subsystems and system design as well as integrating manufacturing constrains in the computer aided modeling. In terms of analysis, companies usually rely on their in-house, proprietary products. While the use of external products such as Ansys or Abaqus is not completely excluded, the in-house tools are essential because the structure analysis inherently requires many different types of calculations which vary from one company to the other. These tools contain stress analysis methods that examine the effect of loads on the different structures and their components. The difference between these tools can be in the types of analysis included, the tolerances, the output format or even the way it adapts to the component under study. At the end, the purpose is in most cases the same: the analysis has to point the regions of failure or improvement.

In general, during design operations, the engineer is more concerned with manufacturing constraints, while loads and weight reduction are in the center of structure analysis. This definitely doesn't mean that manufacturability is not of interest in analysis and weight is not involved in design, but it rather highlights which constraints are prioritized over the others in most cases. This also leads to the conclusion that design and analysis are tightly related and can be regarded as the foundation for the rest of the operations in the detailed design process.

Considering that each discipline has its own tools, it is evident that some extra tasks need to be added to link these two disciplines and channel data between their respective tools. As illustrated in figure 2.3, these tasks are the geometry extraction and the design update. The geometry extraction consists on taking the measurements from the CAD model and using them as input along with other variables to run the stress analysis tools. The design update is the modification of the CAD model based on the stress analysis results to reflect the improvements that can be concluded. While the actual design and analysis exercises are taken care of using software or tools that are relatively fast, the geometry extraction is done manually which makes it longer compared to all the other activities in this loop.

Design and analysis tools and techniques depend on the materials used. Aircraft com-

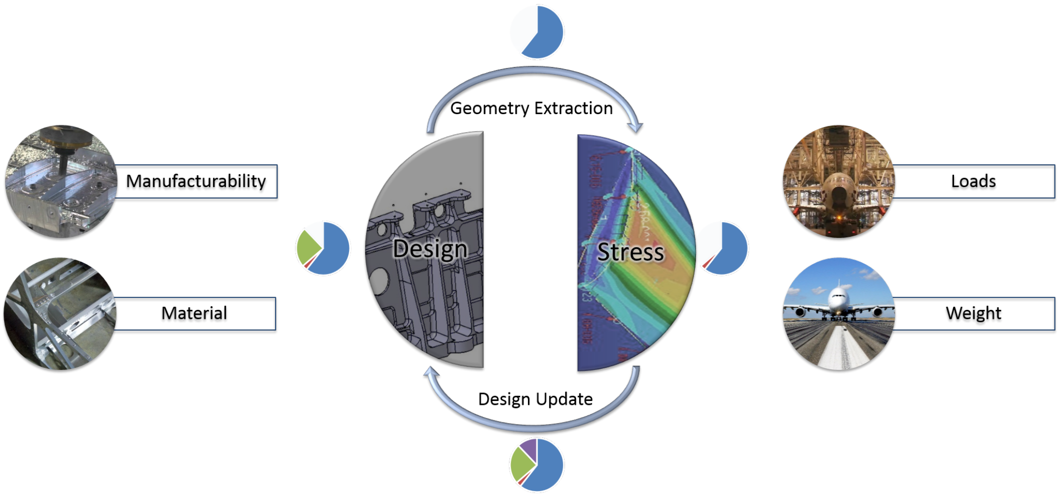


Figure 2.3: Detailed Design Process

ponents are usually made of sheets of metal stiffened by arrangements of stringers[27]. They are complex and redundant which requires simplification or idealization before any analysis can be conducted. Such simplifications occur during the feature extraction phase and sometimes after the results are returned when necessary. Each component is subject to different methods that calculate the deformations, stresses, internal loads etc, all of which should give information about the stability and the vigor of the component to sustain the external loads and forces. Once these results are returned, the engineer has to examine them and locate the areas of failure or improvement. This means that some adjustments need to be made before running the analysis again. Sometimes the changes are only needed in the level of the idealized data which doesn't require an update of the CAD model.

To idealize a component, the engineers usually interact with the CAD model directly to get the measurements either from the history tree or using the measurement tools built in the CAD software. For the type of aircraft this thesis is concerned with, these solids are B-Rep (Boundary Representation) solids which is a method to represent shapes using their boundaries or limits. However, not only this process is long but also many of the parts are imported from old CAD software hence they have no history tree obligating the engineer to take direct measurements from the model. The data collected from the measurements varies from one engineer to the other or between two different measurements. This is mainly due to the fact that the exact definition of an idealized rib might change from one engineer to the other. As an example, in figure 2.4, the widths of the panel can be expressed in two different ways depending on whether the axis used is local or absolute. The difference between the two measurements might not be huge, yet this variability still impacts the stress analysis calculation. Additionally, there is no guarantee that the measurements were taken from the exact same points every time as on a solid with no history there is no indication of how all the different geometries were first created. Also, complicated shapes in B-Rep solids can be confusing in a sense that the boundaries might be divided into small pieces making it difficult to decide which vertex for example or edge is the actual limit of the shape. This inconsistency in the data has an impact on the results generated by the structural analysis tool leading eventually to extra iterations and corrections in order to reach the needed results.

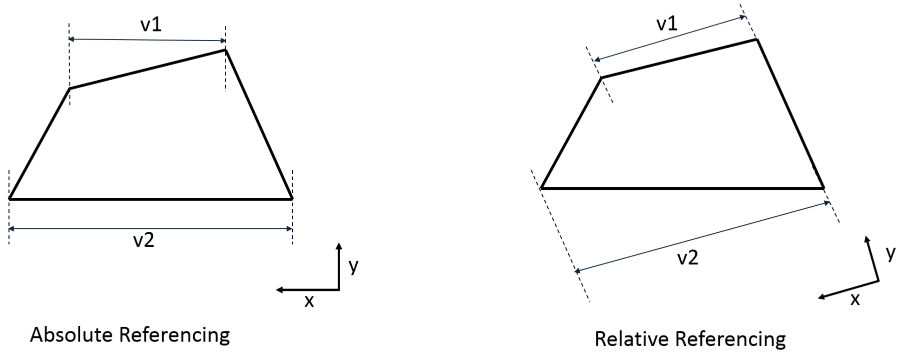


Figure 2.4: Panel Idealization

The engineer's expertise in correcting the data and deciding whether it can be accepted or more tests need to be conducted has a major role in this process. This expertise is of extreme importance in the overall process not only to conclude the analysis and decide what the next step should be, but also as a time consuming operation depending on how much the engineers know about the component under study and the subsequent steps to be taken after the analysis.

Table 2.1: Definition of the Detailed Design Optimization Problem

Decision Variables	⇒	Design variables
Objective	⇒	Reduce weight
Constraints	⇒	Manufacturing, loads, time, procurement and cost

Bringing the big picture of the overall design framework, the processes described above apply to most of the parts on an aircraft. The number of these parts can be as high as few million parts including wiring, bolts, rivets and the different assembly components. In most companies and particularly the large ones, some of these parts are designed and manufactured by external partners from different countries around the world. The ones created internally are usually developed collaboratively between different sites. Additionally, in general this collaboration is alienated with the design process of the part meaning that one site takes care of the conceptual design, another one is responsible for both preliminary and detailed and one more site for manufacturing. Of course this varies from one company to the other, the main idea is that the different major phases of the product cycle are taken care of in different sites, by different teams.

To conclude this section, it is obvious that there are areas of improvement in aircraft design. More specifically, the focus of this thesis is on the detailed design of a component which is highly constrained, multidisciplinary and iterative. If this problem is considered as a pure optimization problem, table 2.1 defines its main elements. Additionally, any improvement needs to account not only for the particularities of the component itself but also of the larger framework in which this component is developed. In the next section, an innovative approach is suggested to tackle some of the problems mentioned above.

2.2 Research Hypothesis and Objectives

The problem formulation in the previous section leads to the deduction of two hypothesis for this thesis. One issue that was repetitively highlighted is the high number of iterations in the detailed design process. An evident solution would be to have some of these iterations automated or even semi-automated to reduce the cycle time. Feedback from subject matter experts indicated that the CAD feature extraction task in Fig.2.3 is the longest. This is due to the idealization operation that takes place during the extraction. The design parameters cannot be directly used as analysis parameters hence they undergo a conversion to meet the requirements of the analysis tool. This conversion has to be consistent with the stress analysis assumptions and the different variables that affect it. A major challenge that is directly linked to this first hypothesis is how to incorporate the engineer's knowledge into the automation as human expertise in design and analysis is the main driver in the decision making process.

The second hypothesis that can be formulated is the idea that using the Digital Thread concept which was introduced in the literature review of the previous section can help integrating disciplines together and allow better predictions to further support the engineer.

From these two hypothesis, the objective of this thesis is to adapt the Digital Thread concept to fit into the detailed design process of a commercial aircraft by leveraging both software and process solutions. The Single Digital Thread Approach to Detailed Design was developed for this purpose and will be further explained in the next chapters.

CHAPTER 3
METHODOLOGY

3.1 STAnDD Approach

STAnDD comes with the idea of threading the different elements of the aircraft detailed design into one single digital model-based approach through a collaborative framework. It incorporates innovative solutions in traditional processes to allow faster evaluation of the product and more informed decisions about its future performance [28]. It also promotes a collaborative environment by bringing more manufacturing and production constraints into earlier phases of design and bridge the gaps between the different tools.

The product cycle is broken down into three different threads: components, subsystems and systems(see Fig.3.1). As mentioned earlier, each thread has its own operations, principles and tools however data exchange needs to be maintained between the three threads in an adequate manner. This exchange must be in a way that the larger scale threads don't overfeed the smaller ones and equivalently the smaller scale ones give enough information that lead to significant improvements in the larger threads.

Such a framework can be build in the most efficient way by appealing to innovative solutions for Data Analytics and Machine Intelligence, Modeling and Simulation and Multidisciplinary Optimization. The interfaces between the different threads and the solutions integrated have to be created in a High Performance Computing Environment to enable fast, robust, and real-time execution of the different tasks and predictions in detailed design. Additionally, the Single Digital Thread concept itself comes with the idea of a centralized data management unit which is responsible for storing and managing the data exchange between the three different threads.

The actual implementation of the framework is laid over a multi-phase plan that adopts

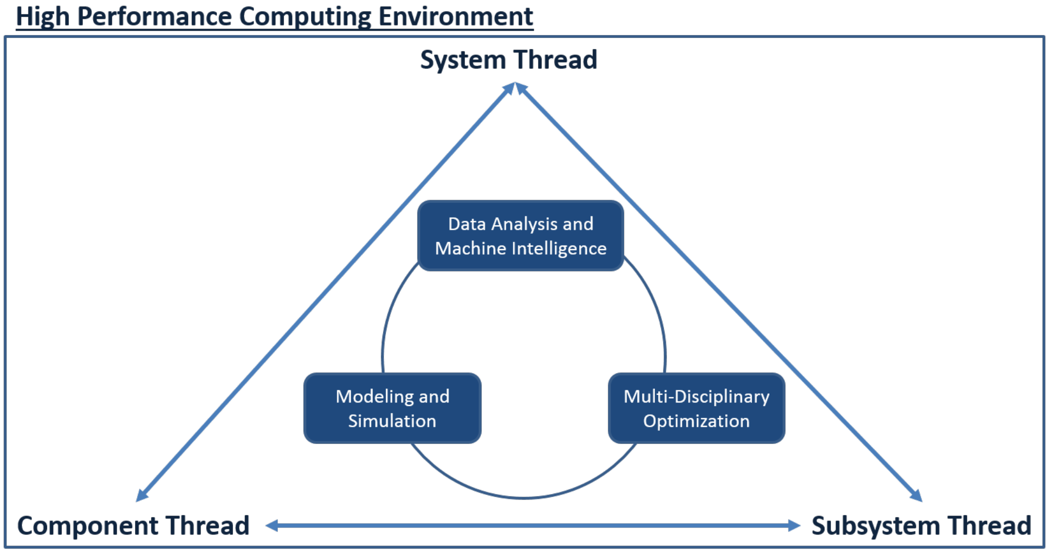

High Performance Computing Environment
System Thread
Data Analysis and Machine Intelligence
Modeling and Simulation
Multi-Disciplinary Optimization
Component Thread
Subsystem Thread

Figure 3.1: STAnDD High Level Architecture

a bottom-up approach illustrated in Fig.3.2. Starting from the foundation of the pyramid, improvements brought to the component thread will be translated up to the subsystems and systems level allowing more and more time savings and potentially earlier predictions. In the component level, the focus is to integrate design and analysis using Machine Intelligence and Simulation capabilities that help reduce cycle time. Multidisciplinary Optimization is then used to support the engineers in the decision-making process which allows a faster and better exploration of the design space with additional time savings. There has to be a clear definition of the optimization objectives here. In the context of structure analysis, optimization aims to reduce an objective function which is usually the weight. However in this case, the optimization framework is meant to improve the decision making process by using the data from the structure analysis results to generate better recommendations for the solid update. A full strategy of how this framework can be built is outside the scope of this thesis and it will be further investigated in future work.

STAnDD is not a new software or a new process. It is rather an improvement of an existing process while leveraging existing software tools with some additional coding. Therefore, STAnDD is innovative in both processes and software. The focus of this thesis is on

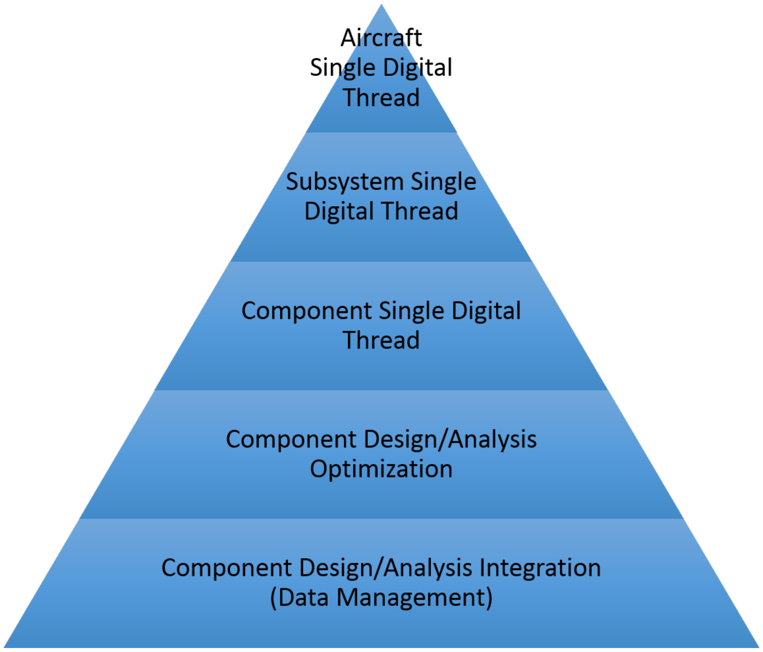


Figure 3.2: Multi-Year Plan

the component thread or the foundation of the pyramid in Fig.3.2. The framework given in Fig.2.1 is the one considered for components in STAnDD and the goal is to reduce cycle time and cost, improve data management and support human decision-making within the predefined constraints.

To demonstrate the capabilities of such an approach, wing ribs were chosen as proof of concept. The reason behind this choice is first because ribs are parts of the wing box which is a crucial subsystem in the aircraft that has to go through a very difficult and meticulous design process. The rib's design process itself is difficult enough to allow the establishment of the foundation work of STAnDD framework but also easy enough to demonstrate the capabilities in a relatively short time hence preparing the ground work for the rest of the wing box components and also the rest of the aircraft. A detailed explanation of the implementation methodology will be given in the next section.

3.2 STAnDD Implementation for Wing Ribs

The way STAnDD will be integrated in this process is explained in Fig.3.3 using the Extended Design Structure Matrix (XDSM) developed by Martins and Lambe [29]. This architecture helps solving multidisciplinary design problems within an optimization framework. In Fig.3.3, each analysis is on the diagonal where their respective inputs are found on the corresponding column. Outputs are found on the corresponding row. Thick gray lines are data flows and thin black lines are process flows.

As shown in the XDSM matrix, the geometric extraction task is a component of the generation of an input file for the optimization task. This process includes a check of the file as to make sure that it can be used by the structural analysis tool. A complete input file has to contain loads data, material database and 2D drawings information. However these data packages don't need any particular treatment compared to the geometric data. Process and data flows are present between the data extraction and all the other analysis boxes in Fig.3.3. This highlights how important it is not only to have a robust and trustworthy methodology to do that but also a quick one. The feature extraction data is tightly linked to the CAD builder task. At this stage, there will be two variations of the rib geometric model: a structural analysis and a design one. The ”structural analysis” rib illustrates an ideal rib whereas the design rib is the real one that is actually built using the CAD software. Having these two variations in hand is extremely useful to guide the decision process and illustrate the compromises between these two disciplines.

To conclude this chapter, and after the explanation of STAnDD approach, the next step is the actual implementation of the foundation of the pyramid in Fig.3.2. This implementation starts with the automatic feature extraction task on a wing rib. The argument behind this start is based on the process architecture given in Fig.3.3. The figure clearly shows a gap between the generation of the input file, the optimization and the CAD builder. Hence the execution of the tasks in this matrix needs to be done in a sequential way starting from

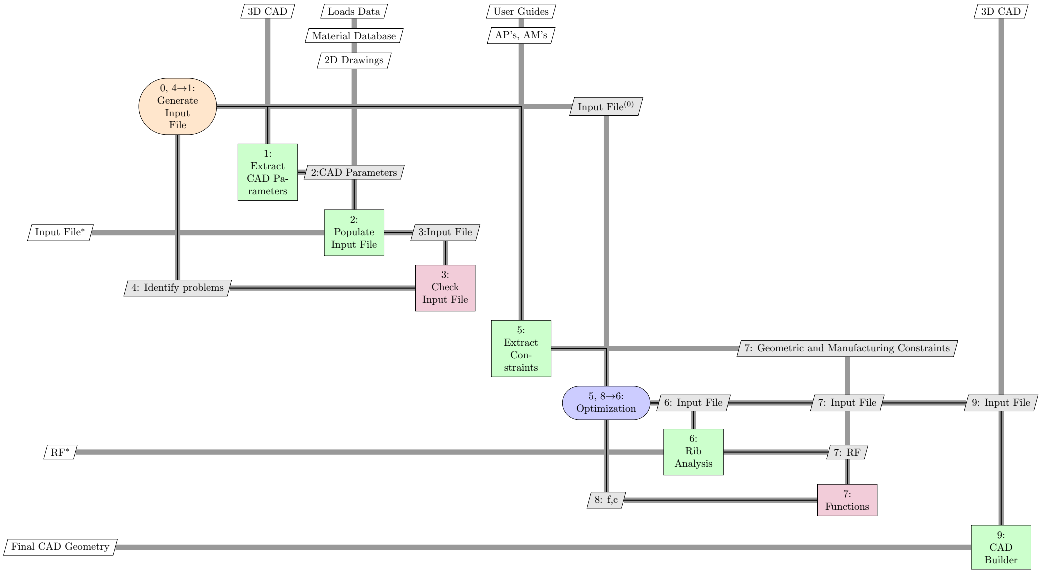


Figure 3.3: XDSM Process

the input file generation.

The CAD model of the wing rib used in the implementation has no history. Measurements taken on this solid are used to populate input files that will later be used in the proprietary structure analysis tools. More details about the automatic feature extraction and the methodology used will be detailed in the next chapter.

3.3 Proof of Concept: The Wing Rib

The rib gives the wing its shape as shown in Fig.3.4. The wing box produces the lift needed to fly. It is composed of main structural components such as the skin, the spars, the stringers and the ribs. It also has many other components such as the tubing, sensors, fuel tanks etc. Therefore, the wing has to sustain a different number of loads including the structural weight loads, the fuel weight, the pressure loads due to the aerodynamic forces and much more, all shown in Fig.3.5.

This puts more emphasis on how important the rib is in a wing box. It captures the airfoil shape of the wing and transfers the loads carried from the skin and the stringers to the spars. Ribs are distributed all over the wing most of the time rearranging the loads on

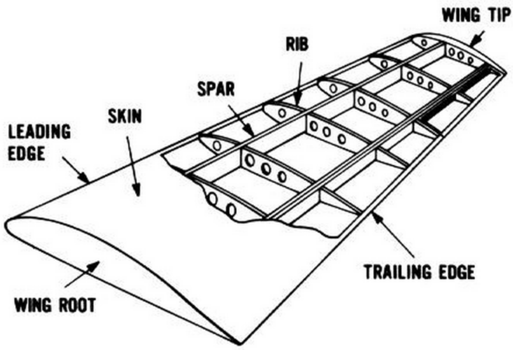


Figure 3.4: Wing Box Elements

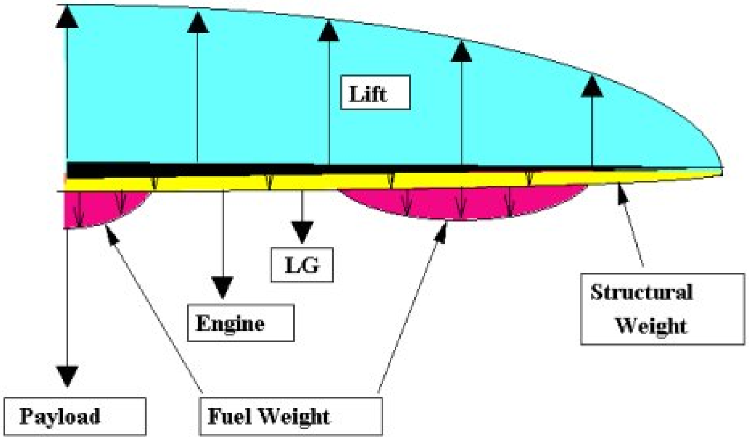


Figure 3.5: Loads Acting on a Wing [30]

the skin panels by breaking them down into smaller pieces and also reducing the length of the stringers. They can also be used to form fuel tanks or to support the tubings and wires of the hydraulic and electrical systems serving different purposes and sustaining different loads based on where they are located. These loads can be internal such as the axial tensions or compressions, the bending moments, the shear and the torsion loads which can either act individually or in a varying combination. Or they can be external such as the crushing, the curvature, the fuel and air pressure loads[31]. Figures 3.6, 3.7 and 3.8 highlight some loads exerted on the ribs in different locations of the wing.

Different types of ribs can be encountered on one single wing. For example, nose ribs are found on the front spar and used to support the leading edge. Bulkhead ribs are on the inboard edge and can be designed to receive compression loads. Fig.3.9 shows some of

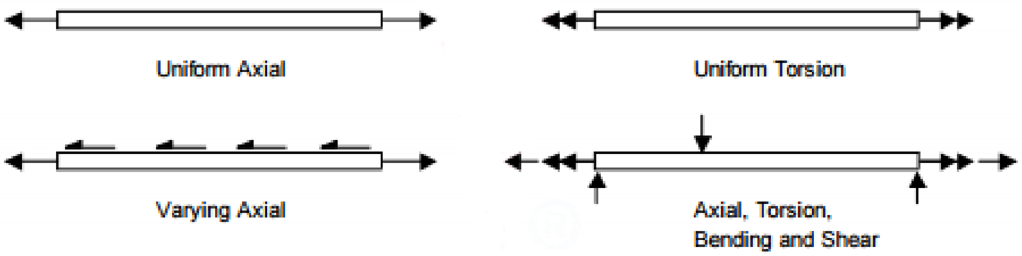


Figure 3.6: Internal Loads Acting on a Rib [31]

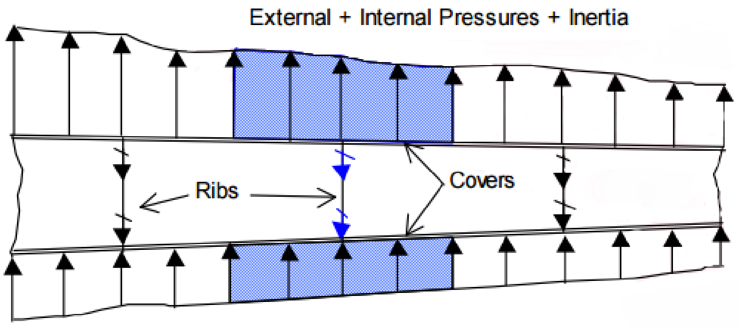


Figure 3.7: Pressure and Inertial Loads Acting on a Rib [31]

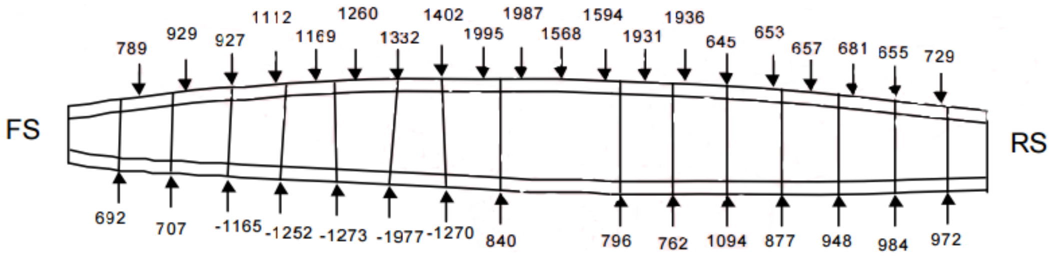


Figure 3.8: External Pressure Loads Acting on a Rib [31]

these types and where they can be found on a wing.

There is a need to take into account the loads applied a particular region, the geometrical dimensions and also the position of the rib in the wing box since this reflects the types of loads that the rib is undergoing and the subsystems integrated(fuel tank, hydraulic tubing, electronic cables etc...).

The complexity of the rib design and analysis process is mainly due to the different features that it contains. As opposed to many types of aircraft, a rib in a commercial aircraft has different features each serving a certain role and even requiring a unique design

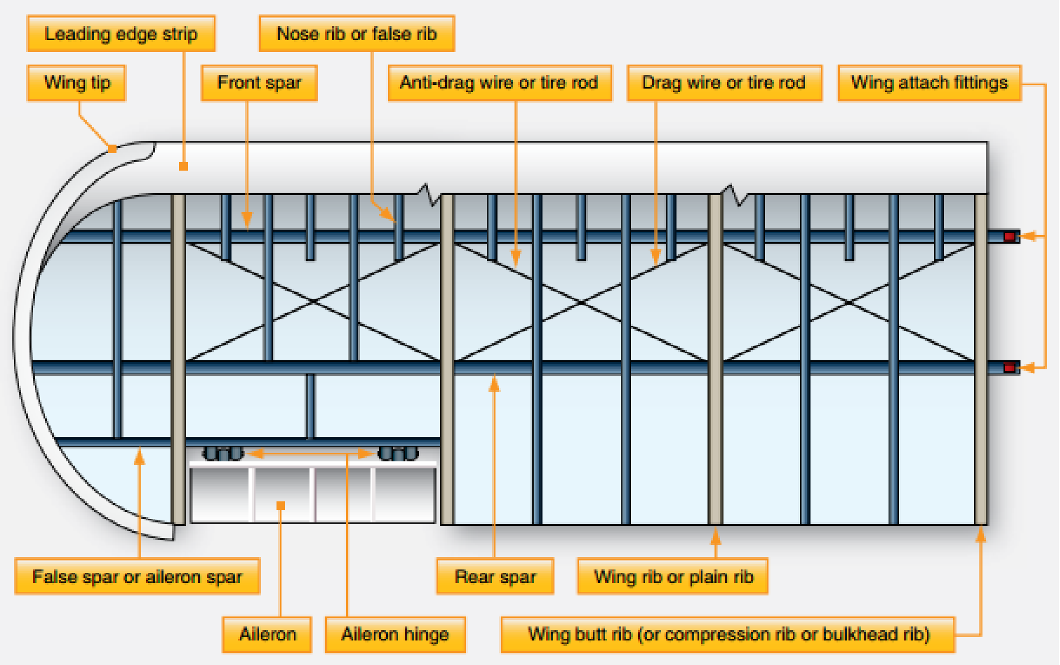


Figure 3.9: Loads Acting on a Wing [32]

and analysis process. In reference to Fig.3.10, a rib can have feet, holes, stiffeners, booms, cleats, seal plates etc. There is always room for some other features depending on many factors however the ones stated above are common to most ribs. Each of these features is further explained below.

Rib Feet The rib feet connect the rib to the skin of the wing and the stringers. They also provide shape to the outer mold line. They use fasteners to transfer loads from the skin.

Holes Holes are penetrations through the rib. They are present to allow for maintenance access, hydraulic and pneumatic piping and electrical cabling to pass through the rib, and also for weight saving.

Stiffeners The stiffeners are the components of the rib which give it strength. The vertical stiffeners support buckling, bending and crushing loads. When the rib requires more than one row of panels across the rib body, horizontal stiffeners are needed to break-up the panels. The horizontal stiffeners support part of the bending and shear loads. The stiffeners can be flat or ramped along their length.

Cleats When the wing flexes during flight, stringers deflect along with the skin. At higher and higher loading, the stringers on each top panel assembly may want to roll over or rotate relative to the skin. If this can be prevented, then the assembly can continue to load up and ultimately be lighter since all the parts are working to a higher level of strain. Cleats serve this purpose by preventing the top skin panel stringers from deflecting laterally during flight when the top stringers are in compression. A cleat is required at every stringer to rib joint, which makes for hundreds of parts per wing.

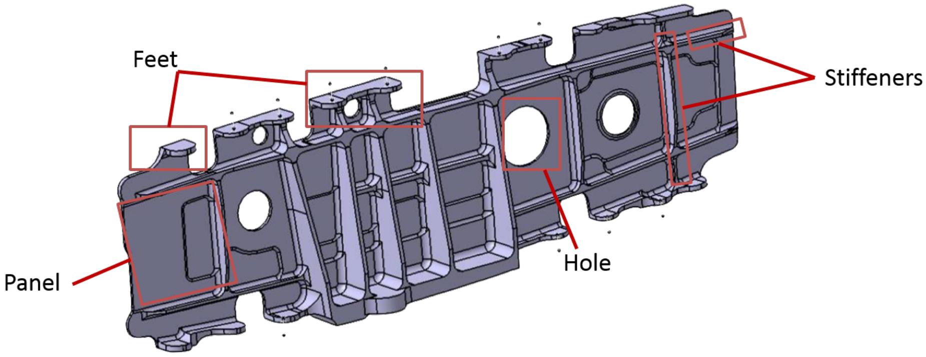


Figure 3.10: Notional Rib Features

In its most basic architecture, the rib can simply be described as sets of panels divided by stiffeners. Features on this simple architecture come in different variations depending on the rib location, the type of loads sustained and the subsystems inside the wing box. The choice of a particular variation depends on the design guidelines of the company and also the engineer's expertise. This suggests that the number of possible configurations is very high. To illustrate that, table3.1 represents the morphological matrix of the different design parameters. This matrix is a tool that provides a structured and systematic way of generating a large number of possibilities[33] and it is used to give an overview of the scale of the problem even for one single component. Despite the fact that this thesis is not concerned with the details of how and why a particular configuration is decided, knowing how large that number highlights the importance of having a framework that can easily

Table 3.1: Morphological Matrix of the Design Task[28]

Feature	**Number in one rib**	**Example**
No of rows	m	3
No of columns	n	14
No of panels	$m \times n$	42
No of vertical stiffeners	$m + 1$	4
No of horizontal stiffeners	$m \times n$	42

Feature configuration							**Combinations**	**Example**
Stiffener section profile	Section 1	Section 2	Section 3				$3 \times [(m \times n) + (m + 1)]$	138
Type of holes	Manhole	Lightening	System				$3 \times (m \times n)$	126
Types of booms	Simple	Double					$2 \times (m + 1)$	16
Castellations	Type 1	Type 2					$2 \times n$	28
Wall	Shape 1	Shape 2					$2 \times (2 \times n)$	56
Configurations	Type 1	Type 2	Type 3	Type 4	Type 5	Type 6	$2 \times 6 \times (2 \times n)$	336
Cleats	Type 1	Type 2	Type 3				$2 \times 3 \times n$	84
Seal plate							$2 \times n$	28

manage and encompass all the possible configurations. More about the objectives of this thesis will be detailed in the next section.

3.4 Methodology to Implement STAnDD in Component Level

Section 3.2 has given a detailed implementation methodology for a wing rib. To generalize this methodology to all components some guidelines must be stated to explain how it was generated.

To implement STAnDD into the detailed design process of any component, there must be a vigor understanding of the tools used in design and analysis as well as the component itself. For that, a survey needs to be done in order to reference and categorize all the variations of the component and its different features. With this categorization, the design principles need to be understood as well for they are the only reference left to understand how a particular design was made since the solids that this thesis is concerned with have no design history.

After a good understanding of the component and the process have been established, a survey of the techniques that can be used for a particular task need to be conducted as well. STAnDD doesn't come with a pre-made list of techniques that will be used everywhere. It rather defines the general path of how an implementation strategy can be created. Recalling that the main objective of STAnDD is to adapt the Digital Thread principles to the detailed design process of any component, system or subsystem, hence each case study need to be taken care individually. The next chapter will give the details of the implementation of the automatic feature extraction for the wing rib.

CHAPTER 4

IMPLEMENTATION: DESCRIPTION OF THE AUTOMATIC FEATURE EXTRACTION PROCESS

As announced in the last chapter, the multi-phase implementation given in Fig.3.2 starts with the automatic feature extraction. The geometry extraction in the detailed design process given in Fig.2.3 is the task of idealizing the geometric data of the features and feeding them into the proprietary structure analysis tool. This task is very long and has so many errors as explained in the previous chapter. The goal is hence to develop a tool that is able to correctly and efficiently perform the idealization operations. This tool will need to interact with a CAD software, which will be CATIA in this case since it is widely used by aircraft companies, and also with the proprietary structure analysis tools.

The tool developed cannot be regarded as an independent one that only performs certain tasks. It is a component that needs to be integrated in STAnDD framework and particularly in the XDSM process given in figure 3.3. The check function of the input file can be fulfilled by the use of visualization techniques and allowing the engineer to interfere to approve or correct the data when needed. This intervention will be kept minimal as the tool needs to be able to operate automatically and make the appropriate decisions when needed. A high level description of the major operations of this tool is given in figure 4.1. The tool needs to operate on a dumb solid in CATIA, identify the different features, perform some measurements and overwrite an I/O file. The development of such a tool can be broken down into five major phases displayed in Fig.4.2 each will be explained in more details in the following sections.

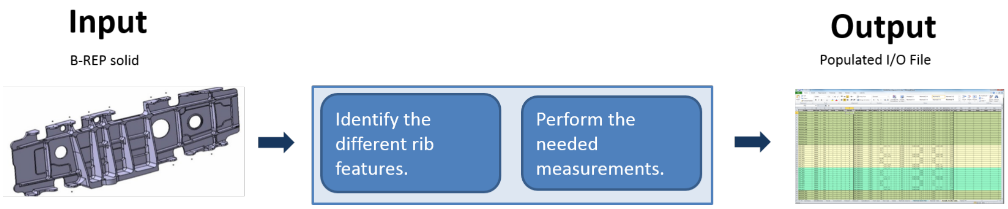

Input
B-REP solid
Identify the different rib features.
Perform the needed measurements.
Output
Populated I/O File

Figure 4.1: High Level Description of the Software

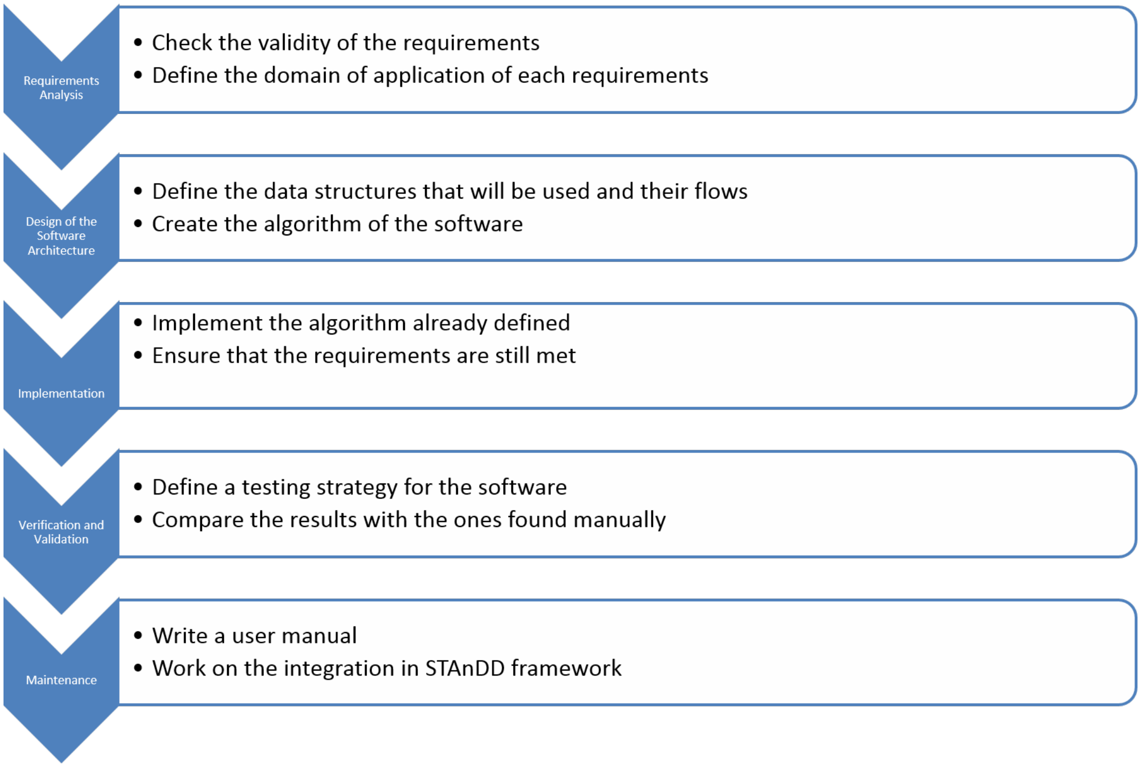

Requirements Analysis
Check the validity of the requirements
Define the domain of application of each requirements
Design of the Software Architecture
Define the data structures that will be used and their flows
Create the algorithm of the software
Implementation
Implement the algorithm already defined
Ensure that the requirements are still met
Verification and Validation
Define a testing strategy for the software
Compare the results with the ones found manually
Maintenance
Write a user manual
Work on the integration in STAnDD framework

Figure 4.2: Process of the Feature Extraction Software Development

4.1 Requirements Analysis

The first phase of the tool development is the requirements analysis. The requirements were established based on the process described in the XDSM matrix and after discussion with subject matter experts in rib design and analysis. The following requirements were created to better guide the functioning of this tool:

1. The tool has to interact with CATIA.
2. The tool has to operate with a B-Rep solid with no history.
3. The tool needs to capture the variations in the rib families and their features.
4. Any feature's definition must be consistent along the different rib families.
5. The idealization methods used must comply with the assumptions in the proprietary stress analysis tool.
6. The data extracted must be formatted according to the requirements of the stress analysis tool.
7. The tool needs to be easy to integrate later on in the STAnDD framework.
8. The tool must allow some engineer interaction while keeping the option of being run automatically.
9. The tool must allow the visualization of the data for verification purposes.
10. The tool must be built in a way that allows it to be run as many times as needed. This can be either triggered by the engineer or automatically.

Going through the list defined above, requirement number 2 indicates the use of a B-Rep solid. A B-Rep solid is a solid that is only represented by its boundaries: vertices, edges and faces. No historical data of how the CAD model will be provided or can even be used. Hence with only these basic elements, the tool has to recognize each feature with all its variations. Requirements 3 and 4 will need to have an efficient verification and validation strategy on their own as it is impossible to ensure that these requirements are met for every single family, considering the large number of rib families in big commercial aircraft companies. Hence ribs that are representative of groups of families will be used. As for the features, a generic catalog has to be created in which the different varieties of a feature are documented with their frequencies and in which cases they mostly appear.

The fifth requirement depends on the engineers expertise and their agreement on a unique set of consistent assumptions within the whole company. This requirement is critical in defining the exact strategies for extracting a particular feature since, depending on which feature is under study at some instances, the way the rest of the features are assumed to be can change significantly. The extraction needs to be implemented efficiently to capture these changes without redundancy.

The easiness to integrate in STAnDD framework suggests that every other component of this framework can be easily connected to this tool. For that, some communication channels need to be built. This suggests the creation of a database in which all the data of the framework is stored. The present requirement is mainly relevant when the choice of what data structures can be used is made. These data structures need to be flexible enough to allow any changes in the future without the need of major modifications in the extraction or any other tool developed.

Lastly, it is important to allow the engineer to interfere when needed. Some visualization tools should be implemented as well as the ability to guide the run of operations when needed. The level of sophistication of these visualization tools depends on their role in the current operation. As stated before, visualization in the context of automatic feature extraction is relevant in the verification task. However, the feature extraction is linked to the CAD builder task in the Extended Design Matrix in Fig.3.3. The data used for visualization will be used to build the idealized rib which will help identifying the areas of improvement in the optimization task.

Each one of the requirements stated above will have an impact on a the tool functioning and development. The use of the tools imposed by the company, the predefinition of input and output format and the way the tool is supposed to communication within the framework all have a direct impact on the choice of the programming language, data structures and which recognition methods can be used. The role of the human in the loop will mainly impact the internal functioning by adding different tests or functionalities for the user.

4.2 Design of the Software Architecture

In the second phase of the process in Fig.4.2 a discussion needs to be carried about the data structures that need to be used along with the design of the algorithm. Again, the argument here is that the tool developed will be integrated in a larger framework hence adequate decisions about data structures need to be made at this stage.

4.2.1 Data Structures

The software needs to be built with the idea that the data extracted can be reused later on in the framework as to create an idealized rib or to build the optimization wrapper. The choice of adequate data structure for this problem needs to be based on six different criteria:

- Nature of the problem
- Performance
- Complexity
- Scalability
- Flexibility
- Integration in a database

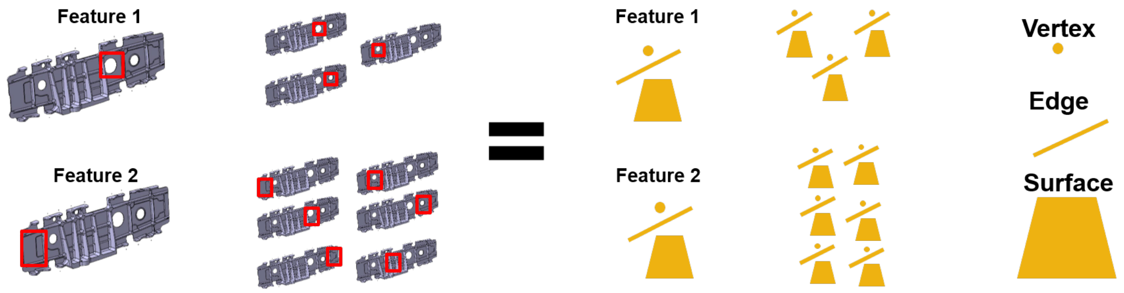


Figure 4.3: Data Structures Map of the Feature Extraction Software

All of the six criteria listed above depend largely on the nature of input and output that need to be managed in this framework. This has already been defined in the requirements phase by imposing the use of B-Rep solids. A feature on such a solid is defined as a collection of edges, surfaces and vertices. The task of the software is to identify the features and their recurrences using the three basic elements on the solid (Fig.4.3).

Performance and complexity are two major criteria for the optimization tool that will be integrated in a later phase in the project. Recalling that one of the major capabilities of STAnDD is allowing fast, real-time decisions and this cannot be done efficiently without a good choice of the data structures used. The scalability is necessary as the number of features on a rib is not constant, therefore the tools needs to be able to handle small or large ribs with a variable number of features. The next phase in the multi-year plan is the implementation of the subsystem thread, this adds up to the scalability of the framework. Flexibility is needed because this tool is built from scratch and has to adapt to the evolution of the tools used in CAD design and structure analysis without major changes. Additionally, the integration of other components later on should be made easy. Finally the integration in a database is a key criteria since STAnDD is built around a central unit responsible for channeling the data between the different threads.

The problem described so far is a good candidate for Object Oriented programming principles. The use of inheritance is an important advantage for code reuse. Inheritance is defined as the capability to inherit commonly used state and behavior from other classes[34], which is extremely practical when dealing with similar features or compo-

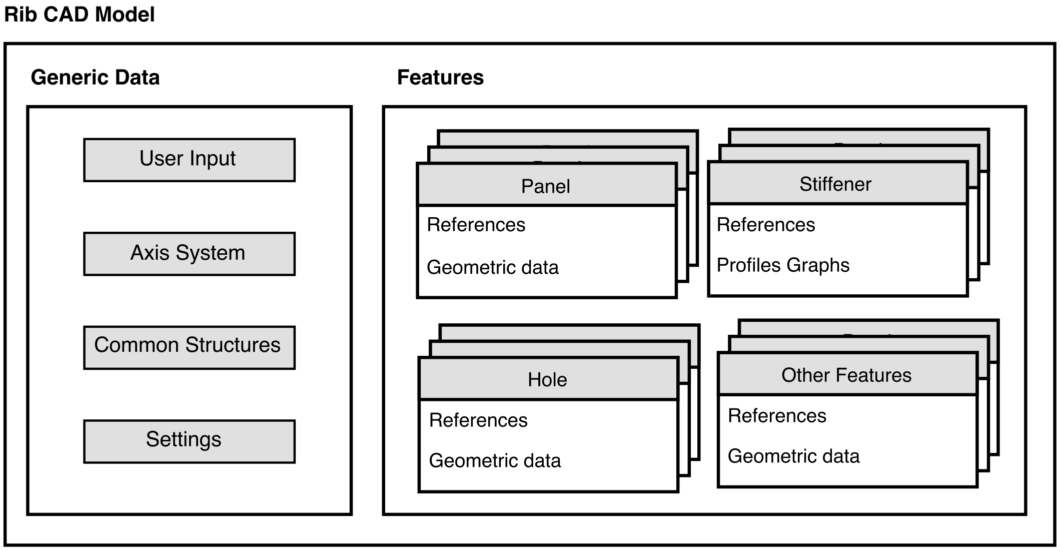


Figure 4.4: Data Structures Map of the Feature Extraction Software

nents. Classes in Object Oriented programming allow the easy addition of variables and procedures. Objects are instantiated from the classes which is useful in the case of recurrences as in the present problem. These instantiations can be stored in lists or tables thus it is possible to store them in databases.

A map of these data structures is given in figure 4.4. For each feature, a new class has been created. These classes were designed in the way that best describes the feature however a common practice was applied. In every class, all the members can be grouped into reference members and geometric data. The reference members are the actual references to the exact structure in the CAD solid. The geometric data are the results from the different measurements or analysis. Some features like the stiffeners require the use of graphs instead since only their profiles are of interest in the stress analysis process. In addition to the features data, some generic data is needed to store the user input or preferences as well as to define the common structures used in the features.

4.2.2 Algorithm of the feature extraction software

The algorithm of the software is provided in Fig.4.5. Only panels, stiffeners and holes are used to illustrate the methods used for feature recognition in this thesis. The I/O file is set up to contain some basic information about the rib that is populated by the user. The reason why some data is entered manually is to add a control or a test from the user to ensure that the software is running correctly. A full automation in such environment cannot be accomplished for now as the engineer has to remain in charge of making some critical decisions in the process until a full automation is certified by the company.

The information read from this file has to describe the rib in a very generic manner and indicate if there are any exceptions in the model under study. This information is used in the software to guide the clustering algorithms and compare what was found with what the user entered. Such information can be the number of rows or columns in the rib for example, or the number of holes.

This information also allows to build the right number of data structures and allocate the memory accordingly. In the present software, every rib feature was attributed to a different class. This approach was chosen because it allows a good management and structuring of the code. It also offers more flexibility to add additional data entries in the future without having to make major changes to the software.

Before the identification task, there has to be a unique way in which the rib or any component can be referenced. The use of absolute axis in the CAD model is not appropriate as this axis is very generic and its referencing changes from one designer to the other. Therefore a local axis system has to be created that imposes a particular way in which the rib can be ”read” or identified. More about this will be explained later.

The next step is the identification of panels, stiffeners and holes. The panels and stiffeners are considered as the basic elements in a rib. Their identification is very important not only for the stress analysis but also for the recognition and localization of the rest of the features. After the recognition operation, the idealization task starts with the application of the appropriate methods to take measurements from a feature. The measurements needed are already defined by the set of input required for the stress analysis and the methods to measure a particular feature is done based on the assumptions of the analysis methods used. In the next subsections, some more details about some important parts of this algorithm will be discussed.

Abstract Definition of Features

The combination of requirements 3, 4 and 5 mentioned in section 4.1 leads to the necessity of having an abstract definition of each feature knowing that the model is a B-Rep solid. From a B-Rep solid, only vertices, edges and faces can be identified hence using only these basic elements and their geometric and spatial dispositions, the tool or the software needs to be able to identify the features represented on the solid.

The geometric and spatial dispositions identifying a feature are called rules in this context. They are created using knowledge of both the rib design procedures and the methods used for stress analysis. In most of the cases, the rules were created in a way that captures a fuzzy feature or a candidate to be feature. Later on, further operations are designed to either reject this candidate or identify its variation. As an example, a candidate panel can be identified as a horizontal surface with respect to a predefined fixed plane. The actual panel has to be bounded by stiffeners. In a first step it is faster to capture all the horizontal surfaces for example then identify the ones surrounded by stiffeners. Later on, more rules need to be imposed in case of the presence of some reinforcements for example or cutouts with reinforcements. This phase translates the human knowledge and expertise into some guidelines that the computer can understand and follow in order to interact with the B-Rep solid.

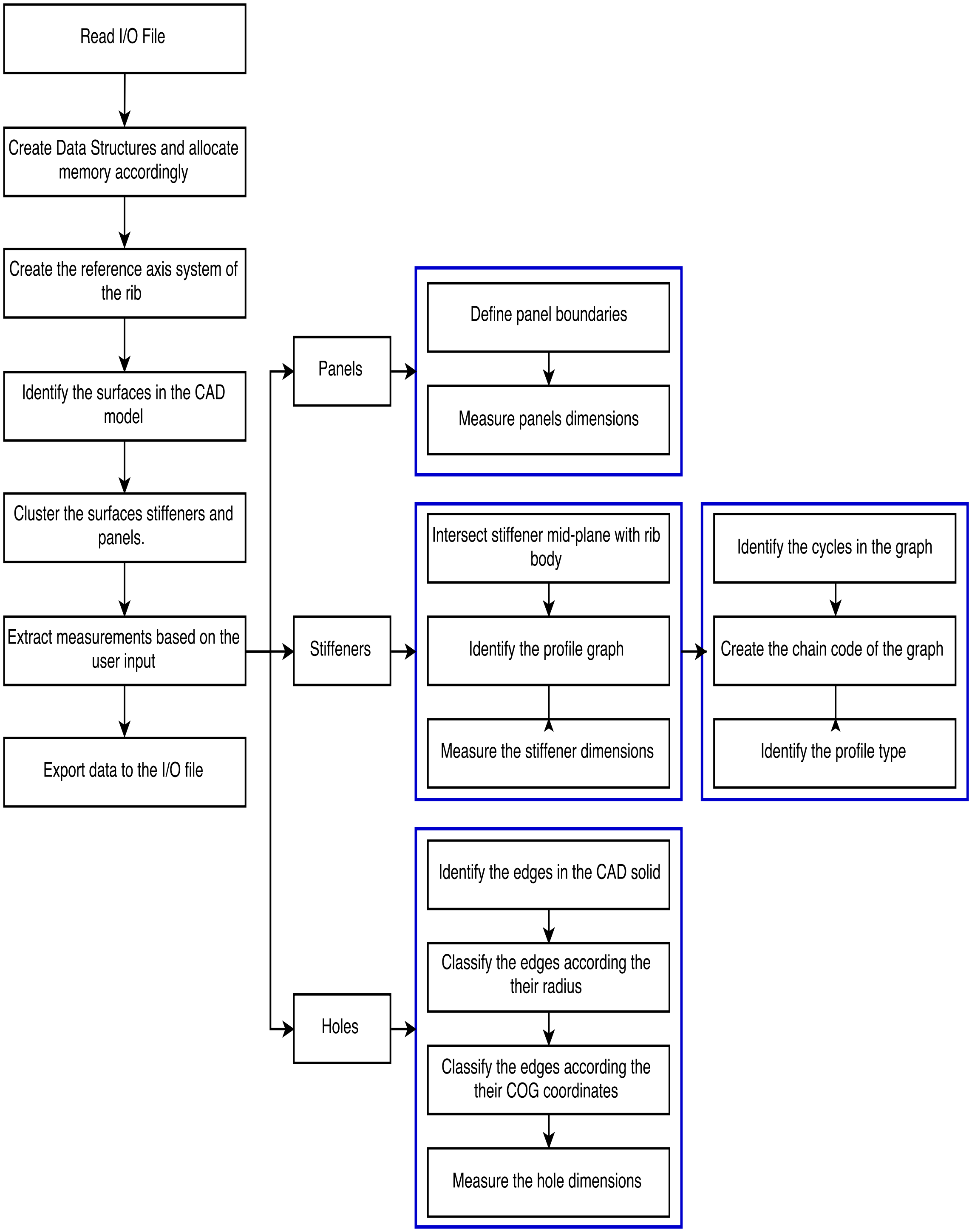

Read I/O File
Create Data Structures and allocate memory accordingly
Create the reference axis system of the rib
Identify the surfaces in the CAD model
Cluster the surfaces stiffeners and panels.
Extract measurements based on the user input
Export data to the I/O file
Panels
Stiffeners
Holes
Define panel boundaries
Measure panels dimensions
Intersect stiffener mid-plane with rib body
Identify the profile graph
Measure the stiffener dimensions
Identify the cycles in the graph
Create the chain code of the graph
Identify the profile type
Identify the edges in the CAD solid
Classify the edges according the their radius
Classify the edges according the their COG coordinates
Measure the hole dimensions

Figure 4.5: Feature Extraction Algorithm

Axis System

Before any extraction can be done, the rib has to be ”read” or referenced uniquely, hence the need of a local axis system. The axis in Fig.4.6 is created from the simple identification of the rib edges and boundaries. For that, the rib location in the wing box must be predefined. Next, the origin is created from the intersection between the front and bottom rib edges. The difference between the front and rear edges or the top and bottom edges is made possible by the use of the information from the location in the wing box. The axis are then created in a way that is most suitable for the extraction of data for the stress tool.The $\vec{y}$ axis is aligned with the front edge. Then a vector $\vec{x'}$ is created pointing from the origin of the axis system to the intersection between the top and rear edges. The vector $\vec{z}$ is generated from the cross product $\vec{z} = \vec{x'} \times \vec{y}$. And finally, $\vec{x}$ is the result from the cross product $\vec{y} \times \vec{z}$. This ensures that the axis system is always orthogonal and right-handed as left-handed axis systems are not accepted in some stress analysis tools. This axis system will allow the rib to be parsed in a unique way, imposing a unique direction from the front edge to the rear edge along the $\vec{x}$ axis , and from the bottom edge to the up edge along the $\vec{y}$ axis.

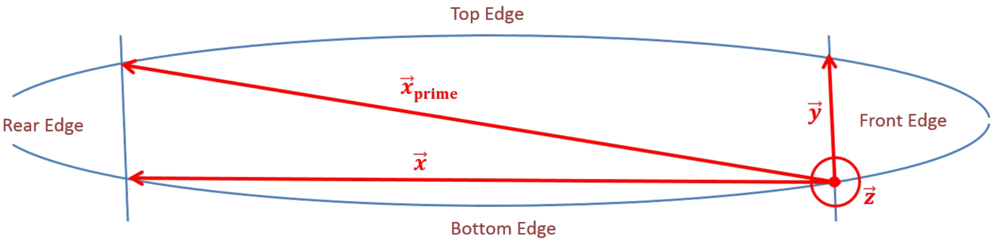


Figure 4.6: Rib Local Axis System

Surfaces Treatment

Taking the most basic rib architecture, there are two features that must be present in every rib: panels and stiffeners. Every panel is bounded by stiffeners except at the edges of the rib sometimes and between every two consecutive stiffeners there has to be a panel except when there is a hole. In a B-Rep solid, these stiffeners and panels are surfaces that are arranged in ways that can be coded in simple rules.

To identify these two features, all the surfaces of the rib have to undergo two classification runs, the first run categorizes the surfaces based on their position with respect to the (xy) plane of the reference axis system and the second run is based on their position with respect to the (xz) plane. The first two steps will create three groups of candidate surfaces to be panels, horizontal and vertical stiffeners. Each group has to go through some extra treatment in order to capture the exact surfaces that compose each feature.

The vertical stiffeners are identified first. For that, the Nearest Neighbor algorithm is used. Algorithm 1 is run on an ordered set of surfaces based on their location with respect to the origin of the reference axis system. The ordering is evidently irrelevant in the Nearest Neighbor algorithm however it is required by the stress analysis tools. In addition to the ordering, a threshold was imposed to exclude surfaces that are too far to be part of a vertical stiffener. This threshold is defined based on the maximum thickness of all vertical stiffeners for a ribs family.

Algorithm 1 Stiffeners Nearest Neighbor

```
1: Start with the first surface as current surface.
2: Find out the closest surfaces to the current surface in the unclassified surfaces set. The distance must not exceed a threshold.
3: Mark these surfaces as classified.
4: If all surfaces in domain are classified, then terminate.
5: Go to step 2.
```

The Nearest Neighbor algorithm is based on finding the closest neighbor to the current surface. For that, an adequate measurement of distance has to be set up. The Euclidean distance and the Manhattan distance were the ones that have been explored. Consider two points A and B given by $A = (x_A, y_A, z_A)$ and $B = (x_B, y_B, z_B)$ in the Cartesian coordinates. The Euclidean distance is defined by:

$$||AB|| = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2} \tag{4.1}$$

Whereas the Manhattan distance is given by:

$$||AB||_1 = |x_B - x_A| + |y_B - y_A| + |z_B - z_A| \tag{4.2}$$

After experimenting with these two distances on a test set, the Manhattan distance has proven to be better in capturing the vertical stiffeners and even the horizontal stiffeners later on. The reason is because in some stiffeners, one side can be broken down into different surfaces while the other is made of a single surface. Since the measurements are made from the center of gravity of each surface, it is more convenient to use a path that is based on the horizontal and vertical directions of the rib rather than the diagonal distance.

The identification of the horizontal stiffeners is done through the same algorithm using the Manhattan distance with the addition of some constraints based on the location of the vertical stiffeners already located. It is important to mention that the Nearest Neighbor algorithm is capable of capturing the horizontal stiffeners without having the vertical stiffeners already defined. However, this method not only helps speeding up the calculation by reducing the set of surfaces on which the algorithm operates but also it allows the ordering of the stiffeners similarly to the vertical stiffeners which needs to be done anyways because in structure analysis, it is important to know which panels are bounded by which stiffeners.

Once the stiffeners are located, the panels are defined as the surfaces parallel to the (xy) plane in Fig.4.6 and confined in a space that is limited by horizontal and vertical stiffeners. If more than one surface is found, the lowest one is chosen to be the panel surface based on the panels rules created for this project.

By the end of this operation, the basic architecture of the rib has been defined which sets up the stage to finding the rest of the features and eventually conducting the necessary measurements to be extracted.

Holes Treatment

Holes are present in the rib for different reasons as mentioned in section 3.3. This implies that there are different number of holes that one can find on a rib. These differences can either be due to the hole diameter, location or the type of features close to it.

Holes are defined using three simple rules:

- Two circular edges are concentric
- The two edges have different heights
- The lowest edges height is 0

This rule also assumes that there are no circular pads on the rib otherwise an extra treatment needs to be done.

Profiles Treatment

In many structure analysis tools, the stiffeners are assumed to be points or simple edges in order to simplify the calculations. However sometimes these assumptions are not sufficient especially when the stiffeners have variable geometries or sustain larger loads. Therefore, using their profiles is a fair compromise between the geometric complexity and the need for simplification for calculation purposes. It is common to assume that stiffeners are symmetrical hence using a midplane between the boundary surfaces is convenient to capture the profile. This midplane needs to be intersected with the solid in order to reveal the actual profile. Knowing that the solid is essentially a B-Rep one, the resulting intersection will be limited to the same basic structures that a B-Rep contains, that is faces, edges and vertices. The faces can be excluded since the use of mid-planes of the boundary surfaces of the stiffeners prevents us from intersecting a surface with itself, therefore no faces can result from this intersection. This leaves us with edges and vertices which happen to be the basic components of a graph data structure.

The graphs in this software were coded in a similar way to the guidelines given in McMillan's book [35]. Two classes were created: a graph class and a vertex class. The edges were represented using an adjacency matrix. The graphs are built so the adjacency matrix reflects whether the edges are straight or curved which is needed later on to identify the shapes of the profiles.

After the profile has been stored into a graph, the shape of the profile needs to be accurately identified. For that, the type of the graph needs to be first identified. Using the family representative ribs, three types of graphs were encountered: Euleurian cycles, open graphs with two leaves and open graphs with four leaves. Any other type of graph is rejected and the user has to do the measurements manually which is one of the limitations of the software. The count of the number of leaves is done through the examination of the degrees of the vertices. If a vertex has a degree 1 then it must be a leaf therefore, looping through all the vertices allows the count of the number of leaves. The Euleurian cycles need to be further examined as according to Euler Theorem, if the number of odd vertices is 0, there is at least one Euler circuit. These circuits are identified using the adjacency matrix. Any vertex can be chosen as a starting point, then a path is constructed moving from one vertex to the other. Knowing that every vertex is connected to exactly two edges, is it safe to always travel through the edge that wasn't already used to get to that vertex. This continues until the starting vertex is re-visited. This operation is repeated until all the vertices in the matrix have been attributed to one cycle.

After the graph type is identified, the shape profile needs to be recognized as well. For that the relative chain codes method was used. The relative chain code is essentially one of the shape recognition techniques for simple objects constructed out of boundaries. Hence in order to find the code of a particular profile, one can imagine themselves walking inside of that shape. Each change of direction is coded uniquely using the numbers in Fig.4.7 leading to an overall unique code for the whole shape.

A relative chain code cannot be created without a start vertex. This start vertex can be

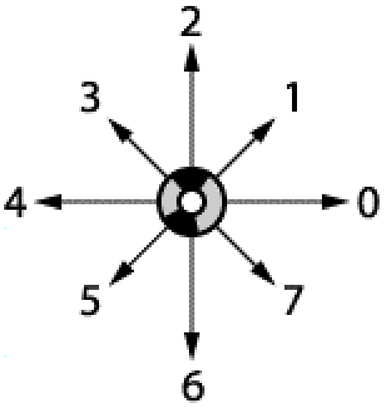


Figure 4.7: Relative Chain Coding Compass

easily found now that the rib has a fixed axis system. The start vertex is found differently for each feature depending on previous knowledge about the feature. For example, the starting vertex of a vertical stiffener can be the one with the lowest y coordinate.

Once this operation is complete, the direction of the profile has to be defined as well. This is important to ensure that all the profiles are coded in the same way. Since the graph used is not directed, there has to be another way to impose a direction for the profile. This is simply done by finding the orientation of the triangle centered in the start vertex. The orientation depends on the signed area given in the equation below for a triangle defined by three vertices $V1(x_1, y_1)$, $V2(x_2, y_2)$ and $V3(x_3, y_3)$.

$$\Delta = \frac{1}{2} \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \tag{4.3}$$

If the area is strictly positive then the triangle is clockwise, else if it is strictly negative then the triangle is counterclockwise as shown in Fig.4.8. For this project, the clockwise direction was imposed for all profiles.

Now that the orientation of the profile is determined, a code has to be generated to describe the shape and where each vertex is located. This again depends on the orientation of the triangle and also the angle at the vertex under study. That is for each vertex, the

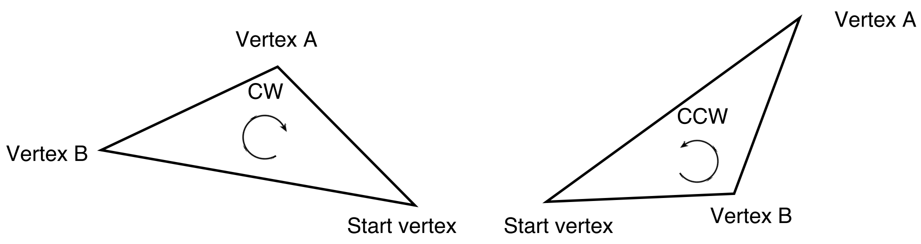


Figure 4.8: Triangle Orientation

previous and next vertices are used to find the orientation of the triangle at the current vertex. Then the cosine and sine of the angle at this vertex are calculated to estimate the angle and consequently generate a code based on all of this information. This is explained more in algorithm 2. In Fig.4.9, two different profiles are given with their respective chain codes. A missing edge is coded as -1 and a circular edge is coded as 10. For each vertex, the previous and next vertices are used even when there is a missing or a circular edge.

Algorithm 2 Relative Chain Code Generation

```
Find the start vertex
Impose a clockwise walk in the shape
for each vertex in the profile graph do
    Determine the orientation of the triangle formed by the triangle at the current vertex
    Calculate the cosine and sine at the current vertex angle
    Assign a code
```

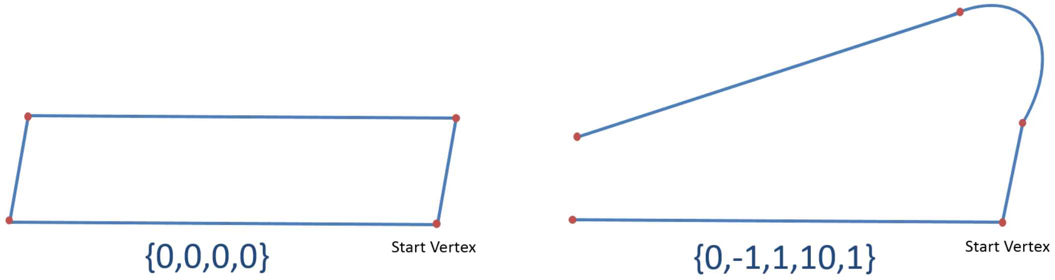


Figure 4.9: Examples of Chain Codes of Different Profiles

For all the profiles that could potentially be encountered on a rib, the method above leads to having an exclusive definition for each one. Therefore, if a new shape is under study, it is sufficient to compare its chain code to the standard ones that are stored in the program. Eventually, new profiles can be simply added by storing their chain codes.

Recognizing a profile is very important in rib analysis as it reveals many information about the stiffener and its surroundings. In the presence of a hole or a reinforcement, there has to be an indication to correctly chose the analysis module and of course return the right results. All of this can be captured in the profile which explains why it is important to have a robust method to manage them.

The graphs where these profiles are stored are coded in a way that allows an easy extraction of the data needed for the analysis. It is therefore possible to measure the distance between two vertices by simply using their respective coordinates. Furthermore, the chain code allows to quickly find a particular location in the profile if further analysis in that region is required. All of this has been also supported by incorporating a visualization of the profiles in the software. This enables the verification process of the operations carried and also helps the engineer have a close investigation of a particular profile when necessary without the need to operate with CATIA and the CAD solid which is time-consuming.

4.3 Implementation

The implementation phase is the actual coding of the algorithm and the operations described previously. For that, an appropriate choice of the programming language need to be made. There are many programming languages that can be used to fulfill the objectives already detailed. Some criteria have to be set though to allow a comparison between all of them. These criteria are:

- Interaction with CATIA
- Ability to overwrite an I/O file for a range of stress analysis tools
- Ability to interact with a database
- Tools offered to build a user interface

Table 4.1: Comparison Between the Candidate Programming Languages

	Python	VB.NET	Knowledgeware	VBA
Interaction with CATIA	Difficult	Easy	Easy	Easy
Ability to overwrite an I/O file	Easy	Easy	Difficult	Easy
Ability to interact with a database	Easy	Easy	Not Possible	Difficult
User interface tools	Rich	Rich	Not Possible	Limited

Four candidate languages have been investigated: Python, VB.NET, VBA and Knowledgeware which is the automation language implemented within the Knowledge Advisor workbench in CATIA. The attributes in table 4.1 were given based on previous experience with these languages and some literature review. From this table, it can be concluded that VB.NET is the most suited language for this task.

VB.NET language have a library to unable the interaction with CATIA. It is also an Object Oriented language that the flexibility to create the data in the way that was earlier described. A very rich Graphical User Interface can be built using VB.NET and the connection to a database is very simple.

4.4 Verification, Validation and Maintenance

Once the software is fully implemented, a testing strategy will have to be designated so as to verify that the software is working properly and achieving the needed results. Part of this strategy has actually been determined in the requirement analysis when decisions were made about the representatives of rib families and the way features and their variations are categorized. The verification and validation will also define the limitations of the software and help set up a maintenance strategy.

There are several Verification and Validation techniques that can be used in this project. The choice depends on the component and the context in which the software is developed. For this thesis, the choice is to find techniques to ensure that the output file is generating the appropriate results. Hence, output files generated by the software were compared to the ones populated by the engineer. Subject matter experts were consulted in the case of discrepancies to either correct the functioning of the software or to affirm that the results are correct. The software was equipped with some visualization tools that allow the engineer to further validate the functioning of the software. It must be noted as well that the software didn't have any random elements in it, therefore, all the runs of the same rib generated the exact same results.

Finally, the maintenance phase is not just about fixing the bugs, a clear and concise user manual has to be written while setting the stage for the integration in the STAnDD framework.

The three first phases of the process in Fig.4.2 have been completed. The last two are currently being run in parallel with the expectation of a full deployment of the software by January 2017.

CHAPTER 5
RESULTS AND CONCLUSION

5.1 Conclusions

STAnDD is an approach developed to improve the existing design processes of commercial aircraft by using single digital model-based design in a collaborative framework. In this thesis, the strategy of how to implement this approach and what innovative techniques can be used have been discussed. STAnDD leverages the use of existing software in aircraft design while using advanced solutions to reduce cost, time and support the decision making process. The full implementation of such an approach takes few years in a large company therefore a bottom-up implementation procedure was suggested starting with the components level and moving up to the subsystems and systems level. This thesis and STAnDD project serve as a pilot effort to demonstrate how the concept of a Single Digital Thread can be implemented in commercial aircraft processes. It brings this concept into the context of detailed design and suggests some practices to guide the implementation of such an approach so as to fulfill its vision.

STAnDD relies on the use of advanced technologies for a better management of the data throughout the detailed design cycle and using it to assess the present aircraft functioning and predict its future performance in a more accurate way.It also promotes the incorporation of human knowledge into the automated operations to reduce cycle time and build up for a single thread framework. Therefore, it appeals to the company's experts to congregate their knowledge into one consistent structure that seeks the best practices in each phase of the design process.

For the application of this approach on a component level, the major implementation tasks have been identified and Chapter.3 was dedicated to explaining the methodology behind the first task which is the automatic feature extraction. The first major asset of the tool developed for this purpose is evidently time reduction. The process of measuring features and extracting them from a B-Rep solid into an external file is demanding and time consuming. Each feature needs to be extracted individually while accounting for any exceptional traits on the rib that require a different treatment. The extraction software not only captures the rib features automatically, but also runs the necessary operations to get the needed information for the structure analysis tool.

This benefit by itself results into two direct gains. First, the software offers a unique and robust treatment of the ribs which excludes any variance related to different assumptions about features or even the errors resulting from the inability to exactly find a particular location on a B-rep solid. Unless a geometric change has been made, every rib will be measured in the exact same way thus remarkably reducing the iterations that have to do with the data extracted from the solid once the analysis results are obtained. This eventually will have an impact on the optimization phase in STAnDD framework and will potentially reduce the number of variables in the problem.

The second direct benefit is that by increasing the level of automation in the design and analysis process, the engineer can now focus on other tasks with the confidence that the feature extraction is working properly. Hence, instead of having to decide which point exactly defines the boundaries of a panel for example, the engineer will rather focus on how this panel can further be optimized and how to improve the current design.

This is not to say that the feature extraction software doesn't support the engineer in this decision making process. As a matter of fact, all the operations were coded taking into account not only the variability of the features but also the assumptions required in structure analysis. Hence, the software offers a way to capture the idealized structure-analysis rib and eventually compare it to the actual CAD model and help the engineer point out the differences and therefore make insightful decisions about where improvements can take place.

On a more general level, and by working closely with experts in rib design and analysis, the software can be thought of as a way to sustain the company's expertise in this domain. This is important for large companies as it allows a better use of the knowledge of the whole team leading to a better growth in a collaborative environment which also falls under the umbrella of STAnDD framework.

Thus, it is evident to conclude that re-usability is highly recommended and that is one thing that the software is able to offer. Considering the huge number of components in an aircraft, the same principles can and will be applied to manage these components and eventually to manage the whole aircraft as envisioned in figure 3.2. Of course this wouldn't be possible if the company doesn't have some standards in which all the processes are well documented.

Even though major aircraft companies are already leading in using Machine Learning techniques in many areas and disciplines, this software allows a deeper integration in a the design process itself which is an extremely complicated. The ultimate goal is to exploit the advantages offered by modern techniques without the need of major changes in the robust processes that already exist and which are impossible to replace. The software created doesn't propose a whole different way of extracting data. It rather uses methods that can capture the human knowledge and turn it into a computer program leading to results that should be at least as good as what a human can do.

5.2 Implications

The concluding remarks imply that the approach developed in this thesis will need to have a case-by-case implementation plan every time, which comes with the nature of the project. The Digital Thread is more of a concept that have been used mostly in the military aircraft industry and only generic definitions and tools were available. There is no fixed form plan of implementation that can be used for every type of aircraft. The goal of this thesis is to seek solutions and methods to demonstrate that this concept will result in huge benefits for the commercial aircraft industry as well and to demonstrate some of its capabilities.

5.3 Future Research

The next steps in this project are the implementation of the rest of the tasks in the component level starting from the multi-disciplinary optimization. Bringing back the detailed design framework described in figure 5.1, it can be regarded as a large optimization problem with thousands of design and analysis variables, many constraints and a complicated solutions space to be explored. With such high dimensionality, adjoint gradient-based methods are the best candidates to offer a solution in a reasonable time frame [28]. The goal of the optimizer is to provide the engineer with fast and dynamic areas of improvement on the rib or any other component. The implementation of the optimization is left to be the subject of another project or Masters thesis.

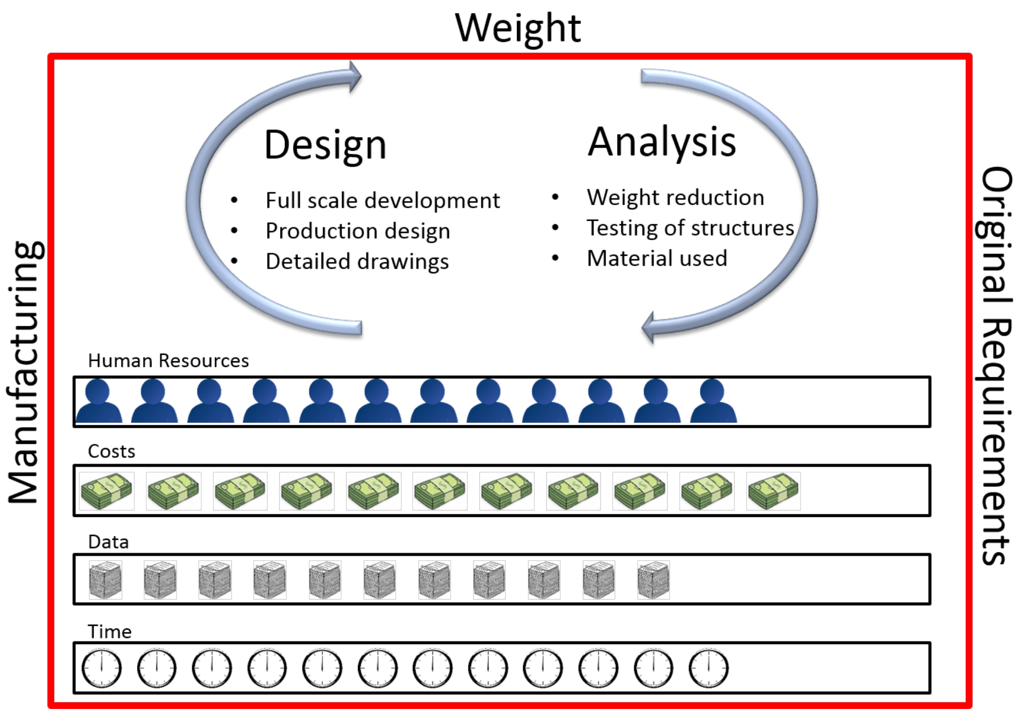


Figure 5.1: Detailed Design Framework

REFERENCES

[1] S. Waters. (2016). Commercial aircraft sketch.

[2] E. Lee-Rausch and M. Park. (2014). Seeking reality in the future of aeronautical simulation, NASA.

[3] W Angela. (2015). Aerospace design, kpucadd.

[4] O. Wyman, Stop the multibillion dollar delays, http://www.oliverwyman.com/, 2014.

[5] D. N. Mavris, “Classical design methods,” Lecture Notes.

[6] NASA, Concurrent engineering guideline for aerospace systems.

[7] M. Cohn, “Collaborative engineering: an approach to modern curtainwall development,” Insight 03, 2016.

[8] A. G. Fernando Mas Jose Luis Menandez and J. Raos, “Collaborative engineering paradigm applied to the aerospace industry,” IFIP, pp. 675–684, 2013.

[9] P. Smith, The waterfall model, 2016.

[10] Airbus, Orders and deliveries, 2016.

[11] Boeing, Historical orders through september 2016, 2016.

[12] E. Kraft, Expanding the digital thread to impact total ownership cost, 2013.

[13] G. Goodwin, Understanding the ”digital thread” in aerospace and defense, 2014.

[14] A. ManTech, Cradle to cradle digital thread for manufacturing, 2014.

[15] E. M.Kraft, “Hpcmp create-av and the air force digital thread,” AIAA SciTech, 2015.

[16] J. T. Brewer, D. N. Mavris, and D. P. Schrage, “Implementation of georgia tech’s concurrent engineering methodology to a high speed civil transport (hsct),” in AIAA, 1994, pp. 731–741.

[17] D. N. Mavris and M. R. Kirby, “Technology identification, evaluation, and selection (ties) for commercial transport aircraft,” in SAWE, 1999.

[18] R. Akerkar, “Analytics on big aviation data: turning data into insights,” IJCSA, vol. 11, pp. 116–127, 2014.

[19] J. K.Williams, “Using random forests to diagnose aviation turbulence,” Machine Learning, 2014.

[20] J. S. Nikunj C.Oza J.Patrick Castle, “Classification of aeronautics system health and safety documents,” IEEE Transactions on Systems, Man, and Cybernetics, Part C, vol. 39, no. 6, pp. 670–680, 2009.

[21] M. Hrastovec and F. Solina, “Machine learning model for aircraft performances,” IEEE, 2014.

[22] A. Variyar, “Application of machine learning to aircrat conceptual design,” 2014.

[23] A. K. P.Arunkumar Anand S.Deshpande, “A system for extracting product ffeature from cad mmodel - a step approach,” Contemporary Engineering Sciences, vol. 3, pp. 139–146, 2008.

[24] M. P.Bhandarkar and R. Nagi, “Step-based feature extraction from step geometry for agile manufacturing,” Elsevier Science, 2000.

[25] J. G. Jan Bohm Claus Brenner and D. Fritsch, “Automated extraction of features from cad models for 3d object recognition,” IAPRS, vol. 33, 2000.

[26] D. P.Raymer, Aircraft Design: A Conceptual Approach, J. A.Schetz, Ed. AIAA Education Series, 2006.

[27] T.H.G.Megson, Aircraft Structure for Engineering Students. Arnold Publishers, 1999.

[28] E. K. D. J. H. V. P. D. R. S. B. D. M. Aroua Gharbi Darshan Sarojini, “Standd: a single digital thread approach for detailed design,” AIAA Journal, 2017.

[29] A. B.Lambe and J. R. Martins, “Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes,” SMO, pp. 273–284, 2012.

[30] A. Web, Structural design, http://www.aerospaceweb.org/design/ucav/structures.shtml.

[31] S. Applegate. (2011). Aircraft structural considerations, Triumph Group, Inc.

[32] Aviation maintenance technician handbook - airframe, Federal Aviation Administration, 2012.

[33] (2005). Morphological matrix, Center for Creative Learning.

[34] Oracle, Inheritance, https://docs.oracle.com/javase/tutorial/java/IandI/subclasses.html.

[35] M. McMillan, DataStructures and Algorithms Using Visual Basic.NET. Cambridge University Press, 2005.