An Architecture for a Multi-Disciplinary Integrated Turbine Rotor System Optimizer

by

Abdulhalim TWAHIR

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE IN PARTIAL FULFILLMENT FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
PH. D.

MONTREAL, MAR 08, 2021

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Hany Moustapha, Thesis Co-supervisor
Department of Mechanical Engineering at École de technologie supérieure

Ms. Sylvie Doré, Thesis Co-supervisor
Department of Mechanical Engineering at École de technologie supérieure

Mr. Amin Chaabane, President of the Board of Examiners
Department of Systems Engineering at École de technologie supérieure

Mr. François Garnier, Member of the Jury
Department of Mechanical Engineering at École de technologie supérieure

Mr. David May, External Evaluator
Siemens

Mr. Pascal Doran, External Evaluator
Pratt & Whitney Canada

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND PUBLIC

MAY 10, 2021

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

FORWARD

Due to the proprietary nature of this research and its potential to give Pratt and Whitney a large industrial advantage, some details have been withheld.

ACKNOWLEDGMENT

Like many who pursue a PhD, at some point I developed a love hate relationship with my work, doubting if I would ever finish or if it was worthwhile to finish. During those dark days I could always count on the support on my supervisors: Professor Hany Moustapha and my co-supervisor Professor Sylvie Doré for that motivation in pushing me through the finish line. I give my sincerest thanks to them for their guidance during my PhD, for being understanding when I had to go part time and was working fulltime at Pratt & Whitney Canada (P&WC). They always looked out for my best interests and made sure I fulfilled my academic requirements. When I got lost and tried to solve multiple problems, they kept me focused and made sure I crossed the line successfully.

At P&WC I would like to thank all the people who helped me, inspired me and challenged me to accomplish all I could. A special thanks to Acher-Igal Abenhaim, Francois Roy and Daniel Lecuyer for helping me understanding what it takes to design a turbine rotor. All I have done is thanks to the knowledge you have imparted on to me. A big thanks also to Benoit Blondin for his help in managing the various projects and getting through all the administrative formalities required when working in a company like P&WC. I would also like to thank Francois Caron, Sami Girgis and Panagiota (Toula) Tsifourdaris for being understanding and trusting me enough to hire me in your departments knowing that I had a PhD to finish. You all gave me invaluable experience working in the Turbine Rotating Structures and Turbine Aerodynamics. Toula especially always had an ear to listen and heart to understand my different goals and for that I am eternally grateful.

This work could not be accomplished without the help of so many students that have been a part of the creation of iRSO. Yanick Ouellet started it all, and along the way Francis Gemme, helped me a lot to define and refine iRSO. Apart from them, I would like to thank all the students for their ideas and dedication to the projects over the years, iRSO is what it because of all of you.

Among all the students I would like to reserve a special thanks to Maxime Moret. Max helped me from the first day, literally showing me the ropes by helping me to get to Pratt & Whitney Canada when I first moved to Montreal. Through the years we have had many ‘fights’ and many jokes (I can think of 4 great ones in particular) and he has always been there for me. Thanks for putting up with me when you had to rewrite your code every time I had a ‘great idea’ that would make the code better but only ended up causing your code to crash whenever I checked in. Thanks for being a great colleague and an awesome friend.

Lastly, to my family both new and old. To my wife, thanks for putting up with me having to work on my PhD and write this thesis when I should have been helping you plan our wedding. Thanks for pushing me to the finish and being with me every step of the way, getting angry at me when I was not writing and needed to. And to the family I was born in, especially my mom and siblings, thank you for making me the man I am today, for always pushing me to achieve my goals and giving me new perspectives.

Cadre pour un système de conception de rotor de turbine intègre

Abdulhalim TWAHIR

RÉSUMÉ

La conception des pièces d'un moteur à turbine telles que le rotor de la turbine implique non seulement la conception de plusieurs composants, mais nécessite également d'incorporer la connaissance de diverse disciplines et applications afin de créer le rotor idéal pour les conditions de conception. Telle une symphonie, chaque discipline doit jouer un rôle et travailler conjointement avec les autres disciplines afin de créer un produit fini optimal. Traditionnellement, la conception de ces composants est séparée entre les phases de conception préliminaire et détaillée. Malheureusement, durant la phase de conception préliminaire, les ingénieurs n'ont pas le temps d'obtenir le rotor parfait et un équilibre délicat doit être atteint entre la fidélité des résultats et le temps pris pour les produire. C'est durant l'étape de conception détaillée que tout l'effort est dirigé sur la précision des analyses et qu'un concept final est atteint. Cette façon traditionnelle de procéder à la conception d'un moteur a une faiblesse: un mauvais concept produit durant la phase de conception préliminaire ne peut pas complètement être réparé durant la phase de conception détaillée, et les tentatives de modification de ce concept afin d'en amenuiser l'impact sur les performances du moteur coûte cher en temps et efforts. L'utilisation de concepts venant de l'Optimisation Multidisciplinaire durant la phase de conception préliminaire (MDO Préliminaire ou PMDO en anglais) permet d'atténuer cette faiblesse en intégrant des connaissances de haut niveau de fidélité durant la phase de conception préliminaire. Cela permet ainsi d'obtenir de meilleurs concepts à la fin de la phase de conception préliminaire. Au sein du programme de Pratt & Whitney Canada (P&WC) sur l'Intégration et l'Optimisation du Système de Propulsion (PSIO en anglais), l'auteur a appliqué les concepts du PMDO afin de créer une architecture pour la conception et l'optimisation des pièces de moteur et a utilisé cette architecture afin de créer un module de conception automatisé et intégré du rotor (iRSO en anglais) ; ce système de conception intègre la conception de la plateforme, du système de fixation, du disque et de la plaque protectrice basée sur les contraintes thermiques et mécaniques, ainsi que la conception de l'ailette basée sur les contraintes aérodynamiques, thermiques et mécaniques et sur les besoins de refroidissement de celle-ci. L'architecture a permis d'injecter plus de connaissance tôt dans le processus de conception, permettant ainsi au concepteur de rapidement développer un rotor complet et d'évaluer ses attributs par rapport à un grand nombre de concepts alternatifs. En partant d'un ensemble de conditions de conception, cette nouvelle architecture permet à un ingénieur d'explorer beaucoup plus de concepts qu'il était possible avant, assurant ainsi qu'un espace de conception très large soit investigué durant les étapes préliminaires du développement et ce avec un degré de fidélité accrus dans chaque discipline. Cette recherche, dû en partie au fait qu'elle se passe dans le monde réel (c'est-à-dire en industrie), si elle avait été conduite en suivant une approche traditionnelle de recherche directe (RD), aurait pu se voir confronter à de multiple obstacles (ce qui a été le cas au début) tels que : ne pas définir clairement des objectifs qui sont compris et acceptés par les chercheurs autant que les utilisateurs finaux, la différence de culture et d'approche de recherche entre l'entreprise et le monde académique, ou encore l'importance de la rigueur académique par rapport au besoin de répondre rapidement aux requêtes des utilisateurs. Ces obstacles ont été adressés par l'utilisation de la Recherche-Action Directe (RAD) qui prend les meilleurs aspects de la RD traditionnelle et de la Recherche-Action (RA) de l'industrie. La recherche dans le domaine des systèmes de l'information / des technologies de l'information (SI/TI) telle que ce travail doit atteindre un double objectif : produire de la connaissance ou apporter une contribution à la discipline tout en aidant à solutionner des problèmes réels avec des paramètres du monde réel. Via la création d'une architecture pour la conception et l'optimisation de pièces de moteur, de la connaissance a été créée quant à la méthodologie à employer pour adresser des problèmes de conception multidisciplinaire ainsi que dans le développement d'outils et de procédures d'optimisation. Comme exemple démontrant les possibilités que le système développé promet, l'optimisation d'une ailette a été accomplie. L'optimisation intègre l'analyse structurelle d'une ailette en 3 dimensions ainsi que ses contraintes aérodynamiques obtenues via analyse CFD. Le système a été capable d'optimiser le concept avec l'objectif double d'augmenter l'efficacité et de réduire la masse et les contraintes maximales à un certain pourcentage de l'envergure de l'ailette, comme le requière les règles de bonne pratique de P&WC. L'optimisation a montré de grande promesse en ne délivrant pas seulement une solution optimale pour les deux objectifs mais également en donnant des renseignements sur le degré de dépendance existant entre certains paramètres clés. Dans ce cas particulier, le système a en effet mis en évidence que la masse peut être diminuée lorsque l'efficacité augmente. Ceci a permis au processus de conception d'aller de l'avant en remettant en question certains acquis tout en réduisant le temps de conception et le nombre de tâches n'apportant pas de valeur dans le travail d'un ingénieur. Cela a également permis de réduire le risque lié à l'approche traditionnelle de la conception d'un moteur à turbine à gaz consistant à avoir deux phases distinctes. Des niveaux de fidélité similaires voir parfois supérieurs ont été atteint en 20% du temps normalement requis grâce à l'utilisation de iRSO. De plus, la possibilité d'erreur humaine est éliminée puisque tous les transferts manuels d'information entre les disciplines sont maintenant contrôlés automatiquement par le système.

Mots-clés: Optimisation Préliminaire et Multidisciplinaire d'un Concept ; Recherche-Action Direct ; Rotors de Turbine ; requis structurels d'une Ailette

An architecture for a multi-disciplinary integrated turbine rotor system optimizer

Abdulhalim TWAHIR

ABSTRACT

The design of gas turbine engine parts such as turbine rotors involves not only designing multiple components but also the incorporation of the knowledge of multiple disciplines and applications to create the ideal rotor for the design conditions. Like a symphony, each discipline has to play a part and work together with other disciplines to create an effective and efficient end product. Traditionally, the design of these components has been separated into the pre-detailed and the detailed design phases. Unfortunately, during the pre-detailed stage of the design, engineers are not afforded enough time to get the perfect rotor thus a delicate balance must then be struck between the fidelity of the results and the time taken to achieve them. In the detailed design however, more emphasis is placed on accuracy of analysis above all else and a final design is achieved. This traditional way of designing can be improved because a suboptimal concept design created in the pre-detailed design step is difficult to correct in the detailed design and attempts to moderate its impact usually come at a steep cost. The use of Multidisciplinary Design Optimization concepts at the pre-detailed design phase (Pre-detailed MDO or PMDO) improves the process by bringing high fidelity knowledge at the pre-detailed stage, allowing for better concepts exiting pre-detailed design. As part of Pratt & Whitney Canada (P&WC) program on Propulsion System Integration and Optimization (PSIO) the author applied concepts of PMDO to create an architecture for the design and optimization of engine parts as well as using that architecture to create an integrated and automated rotor design system (iRSO); a design module that integrated the design of the platform, fixing, disc, and cover plate based on the thermal and mechanical stresses as well as the airfoil based on aerodynamic, thermal and mechanical stresses, and cooling requirements. The architecture allowed more knowledge to be injected into the design process at the early stage of design, allowing the designer to rapidly synthesize a complete rotor and evaluate its attributes over a range of alternative designs. Starting from a set of design conditions, the new architecture allows an engineer to explore many design options, ensuring that a large design space is investigated at the early stages of development with a higher degree of fidelity in all disciplines. This research, due in some part to the fact that it is done in a real world setting (industry), if conducted using the traditional direct research (DR), would have faced obstacle such as: definition of objectives that may not be equally understood and appreciated by the researchers and the end users, the difference in culture and research approaches between the organization and the research institution as well as the importance of academic rigor vs the responsiveness to user requests. These problems were alleviated through the use of Action Design Research (ADR) as it joins the best aspects of traditional DR and the organizational focused Action Research (AR). ADR is helpful as a methodological framework as it recognizes the role of organizational behaviour in shaping the objectives of the research. Research in information systems / technologies (IS/IT) such as this must achieve dual objectives: to build knowledge or a theoretical contribution to the disciple as well as to assist in solving a real world problem in real world settings. Through the creation of an architecture for the design and optimization of engine parts, new methodological knowledge was created on how to best tackle a multidisciplinary design and create optimization capable tools and processes. As an example of the possibilities the system created holds, an optimization of an airfoil was done. The optimization integrated the structural analysis of a 3D airfoil and aerodynamics through CFD. It was able to optimize with the double objective of increasing efficiency and reducing mass and the constraints of peak stress at a certain percentage span range, as per P&WC best practices. The optimization showed great promise in that it did not only give a solution that was the best of both objectives but also provided insight on the degree of dependencies of certain key parameters. For the particular case, it was able to show that the mass could be lowered while increasing efficiencies. It allowed 'the engineering design process [to] move forward by asking "what if" questions and using the answers to make design changes' while reducing the time taken and the non-value added work load on the engineer. This also solved the real world problem of mitigating the risk of the traditional two phased approach to design of gas turbine engines. Similar and sometimes higher levels of fidelity were achieved at 20% of the time when using the artifact. Possibility of human error is also mitigated as all the manual transfer of information between the disciplines is now handled autonomously in the system.

Keywords: Preliminary and Multidisciplinary Design Optimization; Action Design Research; Turbine Rotors; CFD; Airfoil Structural requirements

TABLE OF CONTENTS

Page

INTRODUCTION 1

CHAPTER 1 METHODOLOGICAL FRAMEWORK 17
1.1 Presentation of three frameworks: DR, AR and ADR 17
1.2 Stages and principals of ADR 20

CHAPTER 2 PROBLEM FORMULATION AND LITERATURE REVIEW 25
2.1 Thesis and Design goal 25
2.2 Technological Choices and Literature 27
2.2.1 Multidisciplinary Analysis and Design Environment Approaches........... 27
2.2.2 Geometry Modelling 28
2.2.3 Direct vs Inverse Design Method 29
2.2.4 Approximation Techniques 30
2.2.5 Optimization Algorithms 31
2.3 Objectives 34

CHAPTER 3 BUILD, INTERVENTION, EVALUATION (BIE)................................41
3.1 BIE Schematic 41
3.2 ADR Principal 3: Reciprocal Shaping 43

CHAPTER 4 REFLECTION AND LEARNING 45
4.1 Evolution of Success Criteria..45
4.1.1 Execution Progress.. 50
4.1.2 Engine Station and Cycle Data 51
4.2 Gen.1 Solutions..54
4.2.1 Disc only module 54
4.2.2 Lessons Learnt 55

CHAPTER 5 FINAL ARTIFACT 57
5.1 Coding Practices (req.2-4, req.6-9, req.13, req.14)..64
5.2 CAD – FEA – CAE Software Relationship (req.2-3, req.6-9, req.13, req.14) 71
5.3 CAD Parameterization (req. 5, req. 10-12, req.14)..74
5.3.1 Airfoil.. 75
5.3.2 Platform.. 77
5.3.3 Fixing 78
5.3.4 Disc 81
5.3.5 Cover-plate.. 82
5.4 Multi-Level Fidelity Tools (req.1) 83
5.5 Process Representation with communication between all tools, data, and engineers (req. 5, req. 9) 88
5.6 Airfoil Optimization (req.14)..89

5.6.1 Structural Analysis 89
5.6.2 Aerodynamic Analysis 93
5.6.3 Surrogate Assisted Optimization (SAO) 95
5.6.4 Results 96

CHAPTER 6 FORMALIZE LEARNING 109

CONCLUSION 111

RECOMMENDATIONS 117

LIST OF BIBLIOGRAPHICAL REFERENCES 119

LIST OF TABLES

Table 2-1 Initial (Generation 1) Objectives and Requirement Matrix for Research 37

Table 4-1 Revised Objectives and Requirements Matrix for 2nd Generation of the Software Artifact...47

Table 5-1 User Requirements for both Gen. 1 and Gen. 2 of the Software Artifact 59

LIST OF FIGURES

Page

Figure 0-1 Schematic of a Jet Engine (FAA-8083-3A Fig 14-1) 2

Figure 0-2 Feasibility Sequential Approach (Panchenko, et al., 2002) 5

Figure 0-3 Paradox of Knowledge and Design Freedom in the Design Process (NATO RTO Research Task Group AVT 093, 2006)..6

Figure 0-4 Cost Committed in the Design Process 7

Figure 0-5 Schematic of a Turbine modified from (Groh, 2013) 9

Figure 0-6 Representation of a Rotor...9

Figure 0-7 Disciplines Involved in Creating a Turbine Rotor 10

Figure 0-8 PDDS Schematic 11

Figure 0-9 Engine BOM 13

Figure 0-10 Hierarchy of Software Artifact Nomenclature 14

Figure 1-1 Design Research Methodology (Sein, et al., 2011)...18

Figure 1-2 Action Research Methodology (Sein, et al., 2011) 19

Figure 1-3 Stages of ADR Schematic (Sein, et al., 2011) 21

Figure 2-1 Simple Septic Function 32

Figure 3-1 iRSO BIE schematic 42

Figure 4-1 Initial Design process..50

Figure 4-2 Modified Design Process 50

Figure 4-3 Turbo-shaft/ prop stations 52

Figure 4-4 Turbo-fan Stations...52

Figure 5-1 Hierarchy of Software Artifact nomenclature...57

Figure 5-2 UML - Core Classes....67

Figure 5-3 UML – Component Inheritance....68

Figure 5-4 UML- Turbo machinery Level....69

Figure 5-5 UML - Engine Level....70

Figure 5-6 Architecture for Interaction of iRSO with the Outside World....73

Figure 5-7 Representation of a Rotor....75

Figure 5-8 Airfoil Parametric Model - Section Parameters....76

Figure 5-9 Airfoil Parametric Model - Pocket definition....76

Figure 5-10 Airfoil Parametric Model - 3D Airfoil....77

Figure 5-11 Platform Parametric Model....78

Figure 5-12 Fixing Parametric Model - 1 to 4 Lobes....79

Figure 5-13 Fixing Parametric Model - Neck and Lobe Pair....79

Figure 5-14 Fixing Parametric Model – Unload line....80

Figure 5-15 Fixing Parametric model - Cooling Passages....81

Figure 5-16 Disc Parametric Model....82

Figure 5-17 Cover-plate Parametric Model....83

Figure 5-18 Representation of the CC-FSA....84

Figure 5-19 Multi-Disciplinary Optimization Loop....84

Figure 5-20 Schematic Representation of Multi-Level Fidelity....85

Figure 5-21 Schematic Representation of Multi-Level Targets....86

Figure 5-22 Comparison of Structural Results....87

Figure 5-23 Comparison of CFD results....88

Figure 5-24 An example of GUI of a component tool in the iRSO Module....89

Figure 5-25 Airfoil Structural Analysis – Area 90

Figure 5-26 Calculation of Section Volume 91

Figure 5-27 Representation of Coating Thicknesses 92

Figure 5-28 Example of Structural Results of Airfoils (dummy airfoil) 93

Figure 5-29 Example of Airfoil Mesh 94

Figure 5-30 Example of a Turbine CFD Setup 94

Figure 5-31 SAO Workflow (Doran, et al., 2018) 96

Figure 5-32 Airfoil Optimization Results - Weight Sensitivity 99

Figure 5-33 Airfoil Optimization Results - Max Stress Sensitivity 100

Figure 5-34 Airfoil Optimization Results - Max Stress Location Sensitivity 101

Figure 5-35 Airfoil Optimization Results – Efficiency Sensitivity 102

Figure 5-36 Airfoil Optimization Results – Reaction Sensitivity...103

Figure 5-37 Airfoil Optimization Results - Efficiency vs Reaction 104

Figure 5-38 Airfoil Optimization Results - Efficiency vs Weight...105

Figure 5-39 Automated Fixing FEA 106

Figure 5-40 Disc Fixing Optimized Using iRSO – 3 Lobe (Mavroudi, et al., 2019) 107

Figure 5-41 Blade Fixing Optimized Using iRSO – 3 Lobe (Mavroudi, et al., 2019) 107

Figure 5-42 Disc Fixing Optimized Using iRSO – 2 Lobe (Mavroudi, et al., 2019) 108

Figure 5-43 Blade Fixing Optimized Using iRSO – 2 Lobe (Mavroudi, et al., 2019) 108

LIST OF ABBREVIATIONS

2D/3D	Two/Three dimensional
AR	Action Research
ADR	Action Design Research
BOM	Build of Material
CAD	Computer-aided design
CAE	Computer-aided engineering
CAx	Computer-aided technologies
CC-FSA	Component Centric FSA
D&A	Design & Analysis
DR	Design Research
EDS	Engine Data System
FEA	Finite Element Analysis
FSA	Feasibility Sequential Approach
GUI	Graphical User Interface
iRSO	Integrated Rotor System Optimizer
IS	Information System
IT	Information Technology
LED	Leading Edge Diameter
LEMA	Leading Edge Metal Angle
LEWA	Leading Edge Wedge Angle
MDO	Multidisciplinary design optimization

NATO	North Atlantic Treaty Organization
OOP	Object oriented programming
PDDS	Pre-Detailed Design System
PMDO	Preliminary and multidisciplinary design optimization
P&WC	Pratt & Whitney Canada
TED	Trailing Edge Diameter
TEMA	Trailing Edge Metal Angle
TEWA	Trailing Edge Wedge Angle
UML	Unified Modeling Language

LIST OF SYMBOLS AND UNITS

σ	Stress
ρ	Density
m	Mass
V	Volume

INTRODUCTION

The design of a gas turbine engine is technically difficult and is a highly complex process as it involves the design of five major components (intake, compressors, combustors, turbine, and exhaust) each difficult in its own right. Each of these major components need to be intricately interwoven and be delicately balanced into an efficient, structural sound and commercially viable engine. A team of specialists from multiple disciplines participate in this process and have to navigate and share information throughout the process in order to arrive at a complete product. Like a symphony, each discipline has to play a part and communicate with other disciplines in order to result in an effective and efficient end product.

Looking closely at the major sections of the gas turbine engine, as shown in Figure 0-1, the cold end and hot end each have their unique challenges. The cold end, which is comprised of the intake and compressor, is responsible of taking air at ambient (area of low pressure and temperature) delivering it at a much higher pressure to the combustion chamber. This “compression of air is an unnatural activity. It has been likened to trying to sweep water up a hill” (Rolls Royce, 2015) in its arduousness.

The hot end, comprising of the combustion chamber, the turbines and ending with the exhaust, is conversely responsible for the injection of energy to the flow by adding fuel and burning it to increase the temperature before the extraction of energy from this hot and high pressure gas. The turbines are complex, “glowing red-hot, the blades operate in temperatures well above their melting point; each blade is being stretched by 18 tonnes of centrifugal force as it travels at 500 meters per second” (Rolls Royce, 2015).

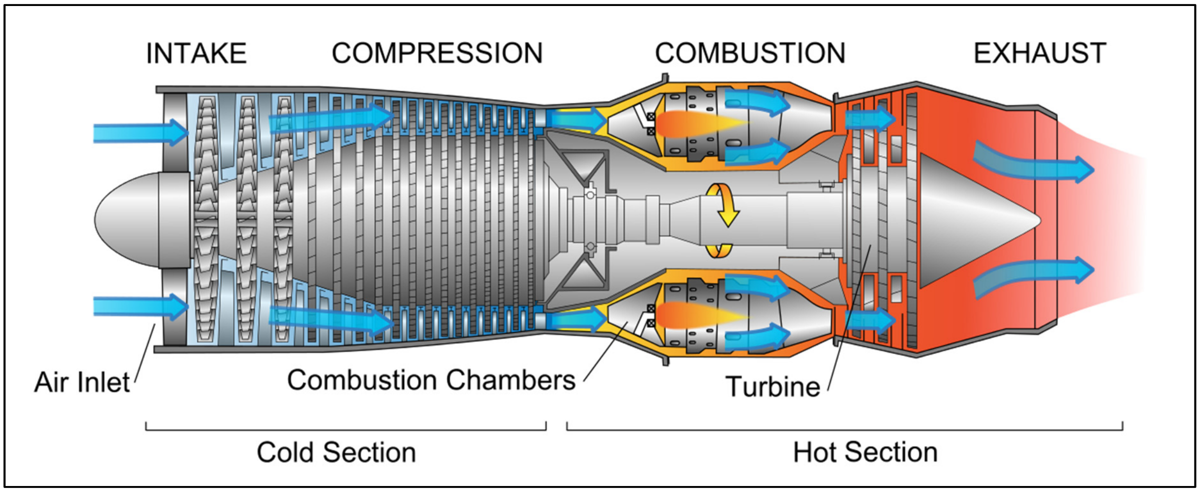


Figure 0-1 Schematic of a Jet Engine (FAA-8083-3A Fig 14-1)

The process of interweaving the relationships between the multiple disciplines (structural, thermal, aerodynamics, manufacturing, etc.) involved in the design of a gas turbine engine often entails the balancing of conflicting requirements; in the turbine, the pursuit of higher efficiencies and lower fuel consumption is often at odds with structural integrity and life requirement. An optimum solution is reached "for each application by performing a series of trade studies that consider all the leading attributes and requirements of the system including life-cycle cost, weight, performance and noise." (Rolls Royce, 2015). On the business side, the design of aero-engine is a race in which to the winner go the spoils (and the contracts), with each company trying to provide a competitive engine in the shortest amount of time. A company has to be fast at responding when airframe manufacturers issue request for proposals (RFPs) and have to be first to market with a product that delivers on the promises made on performance, durability, and safety.

With the pressure of producing better engines faster, the design of gas turbine engines is divided into two distinct phases: pre-detailed and detailed design (Moret, et al., 2017) . During the pre-detailed (preliminary) design phase, the main focus is on creating design concepts as quickly as possible and usually involves iterating between many permutations of engine architecture concepts. Due to the high-level nature of these permutations of engine architecture studies, generalist (engineers with a large breadth across multiple disciplines but not necessarily large depth in any one discipline) are usually responsible for these studies. Lower accuracy but faster methods such as established rules of thumb are used to arrive at a rough idea of the geometry of components, sub-assemblies, assemblies and finally the product. Furthermore, since the aerospace industry is highly specific in the functionalities expected for its tools, most of the tools used at the pre-detailed design stage are in-house programs that translate proprietary knowledge into software owned by particular disciplines. These programs exist in isolated environments with few, if any, links between them.

The detailed design phase is focused on delivering a final product and is done by specialists (engineers with a large depth in one discipline). Over multiple cycles, the engine is molded into its final shape. This phase of design is solely focused on creating an engine based on the architectural concept defined at the end of the pre-detailed phase and through multidisciplinary optimization (MDO). Over the last 40 years or so, significant improvements in computers and computer aided technology (CAx) have revolutionized the detailed design phase of design. They provide better prediction of stress of a part through Finite Element Analysis (FEA) and improvement in engine performance prediction due to better Computational Fluid Dynamics (CFD) modeling. The improvements have led to less testing time, lower cost and the reduction of the risk of overdesigning, consequently reducing the weight of the engine as the accuracy of analysis increased.

This revolution did, unfortunately, come with unintended consequences. The more accurate and more complex tools used in detailed design led (in detailed design) to an increasing number of specialists and increasingly more specialized departments within companies. Segmenting the Design and Analysis (D&A) process into smaller and smaller departments with an increasingly narrow and limited mandate led to the "serial mode of engineering design" (NATO RTO Research Task Group AVT 093, 2006). This approach that accommodated the specialists' understandable need to have autonomy and authority in decision making in their area of expertise led to multiple tools and methodologies being used for sometimes very similar jobs. The isolation of processes, sometimes caused by geographical location differences, meant that there was little communication between the different specialists. The communication between the varied tools used was also lacking as little incentive existed for the specialist to create tools with compatible interfaces, leading to highly coupled complex systems being designed in isolation.

The use of more specialists' tools led to a much higher level of fidelity en route to a more 'optimized' design within a discipline; however, it does fail to efficiently (using the least amount of resources i.e., human resources and time) and effectively (using the best tools for this multi-discipline problem) provide a globally optimized solution. Large amounts of data are generated in these higher fidelity analyses which are not always of the same format and are thus incompatible. A lot of human capital is thus expended doing non-value added tasks such as translation of data from the outputs of one tool to the inputs of another.

This design process has a major disadvantage; sometimes a complete restart of the design process might be needed in the detailed design as "the best engineering effort cannot totally right a poor concept selection" (Panchenko, et al., 2002). With the pre-detailed design phase having such an important impact on the architecture of a new engine definition, as it sets the technical envelope of the product (not to mention greatly affecting the product performance, weight and cost), this process mandates large investments in the detailed design phase, a phase that has a high cost of change, to further refine the very rough preliminary design.

Although different tools are used in each phase of design and different levels of fidelity achieved, they both (pre-detailed and detailed phase) subscribe to the same methodology. Both phases use a Feasible Sequential Approach (FSA) in which the feasibility of a system is assessed by one discipline then the other and other. Figure 0-2 shows a typical example of FSA during the design a turbine as presented by Panchenko et al. (Panchenko, et al., 2002).

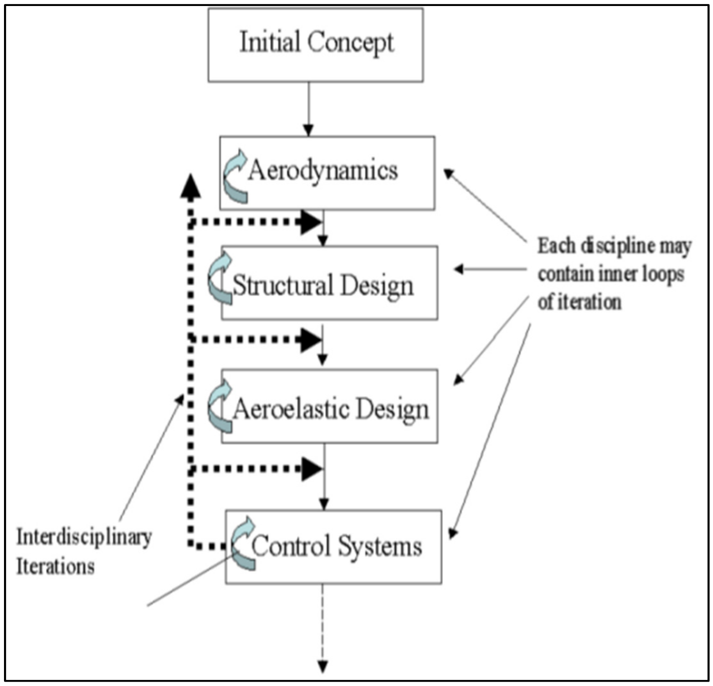


Figure 0-2 Feasibility Sequential Approach
(Panchenko, et al., 2002)

As mentioned by Panchenko et al (2002), the sequential decision approach leads to more data being available as the process goes on while the freedom to change the design in part or as a whole decreases. This Paradox means that when you have all the data to make an informed decision, you no longer have the freedom to act on it. This is shown in Figure 0-3, where the percentage of the total design variables that could be altered and the knowledge necessary to make a decision on design variables is plotted against the time into the design process.

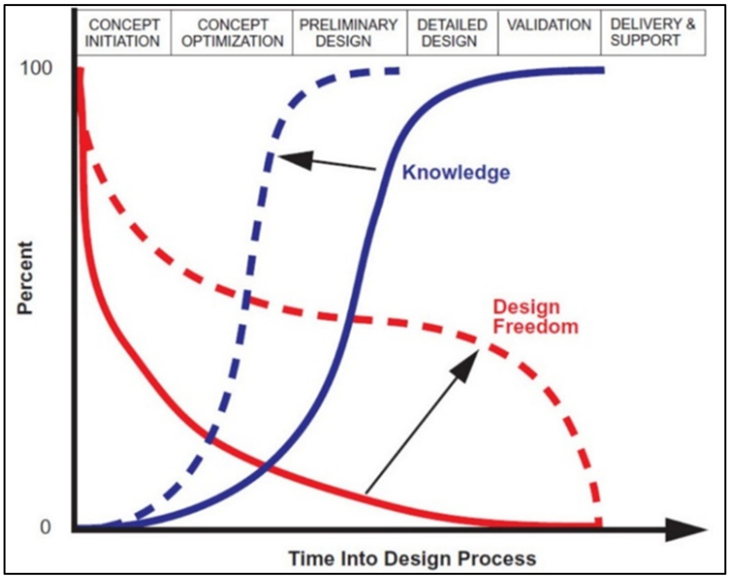


Figure 0-3 Paradox of Knowledge and Design Freedom in the Design Process (NATO RTO Research Task Group AVT 093, 2006)

As can be seen in Figure 0-3, the knowledge available at the pre-detailed (preliminary) design stage is very low and increases almost exponentially with the most knowledge available toward the end of the detailed design stage. At this stage however almost all of the design variables are frozen, and a large amount of the total cost of the system has already been committed and thus the design freedom is at a minimum. This is shown by the solid line in Figure 0-4.

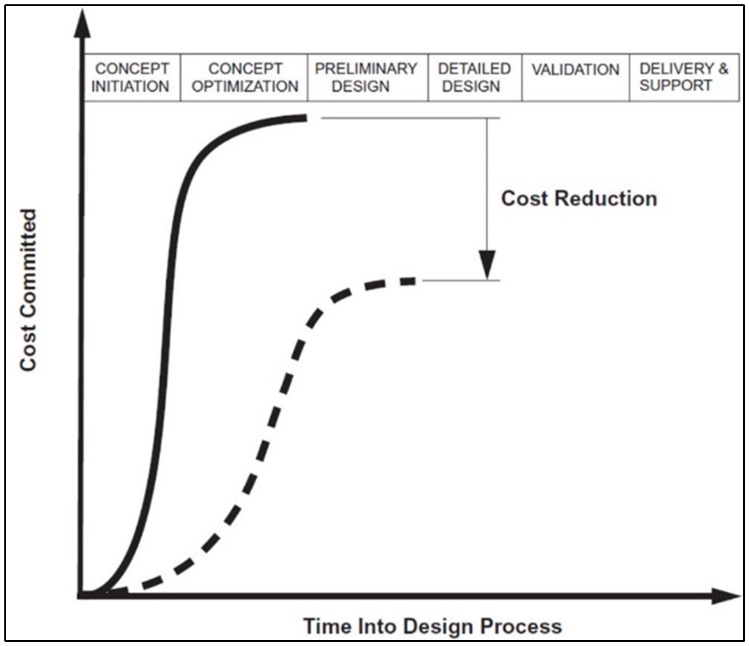


Figure 0-4 Cost Committed in the Design Process

To capitalize on the low cost of change in the pre-detailed (preliminary) design phase, harmonious interaction and integration of all disciplines needs to happen at the pre-detailed stage with higher level fidelity available. This would allow the solid line of the knowledge to be moved back, as shown in Figure 0-3, increasing the design freedom as shown by the red dotted line. The incorporating multi-disciplinary knowledge at the preliminary design phase would mean that better decisions are made earlier on and therefore resulting in the reduction in cost (as shown in Figure 0-4) and a reduction in the risk brought on by the aforementioned paradox.

Studying the current process, it becomes clear that to achieve the harmonious interaction and integration of all disciplines at the pre-detailed stage, specialist tools and knowledge traditionally used at the detailed stage would need to be utilized but it is also clear that these tools cannot be used as is since they are traditionally slow. Multidisciplinary optimization (MDO), a solution that is optimized for a variety of disciplines congruently, would need to be adapted for the pre-detailed phase so that the speed of execution (ability to evaluate multiple concepts) is not lost whilst leveraging higher fidelity tools. This adaptation would lead to the

Multidisciplinary optimization at the pre-detailed stage (pre-detailed MDO or PMDO), a new process as applied to various components and assemblies.

PMDO is especially critical to Pratt & Whitney Canada (P&WC). As a company with "fourteen distinct engine families: JT15D, PT6A, PT6B, PT6C, PT6T, PW100, PW150, PW200, PW210, PW300, PW500, PW600, PW800, and APUs" (Pratt & Whitney Canada, 2019) and "63,000 engines in service" with "13,400 customers" (Pratt & Whitney Canada, 2019), to maintain its status as a world leader it routinely creates 30-50 engine concepts a year. With the high cost of the detailed design stage, sub-optimum choices of design parameters made during pre-detailed design would be multiplied by 100 at P&WC as they have "introduced 100 new engines powering a variety of aircraft in the past 25 years" (Pratt & Whitney Canada, 2019). Its large portfolio of engines and the many concept studies done each year, constantly responding to RFP or trying to stay ahead of the game, P&WC needs to constantly improve its pre-detailed design tools and processes.

To apply the tenets of (PMDO) and improve upon the conventional sequential discipline feasibility approach, a research chair was created between industry, P&WC, and academia, École de technologie supérieure (ÉTS), with the goal of deploying a fully automated and integrated propulsion system optimizer (called PDDS, pre-detailed design system). The creation of an integrated propulsion system optimizer would be a moon shot and therefore the research team decided to deploy the system in multiple phases with phase one focusing on the turbine. This thesis presents a large portion of this first phase, presenting the architecture created for PDDS for the PMDO of the turbine rotor.

To quote the man who set humanity on its first moon shot, PDDS decided to focus on the turbines "not because they are easy, but because they are hard" (JFK Moon Speech). A turbine consists of one or more stages of rotors and stators (Figure 0-5). The stationary vane (stator) acts as a nozzle dictating the gas the blade sees while the rotating blade (rotor – yellow coloured) extracts thermal energy and converts it to kinetic energy through its rotations.

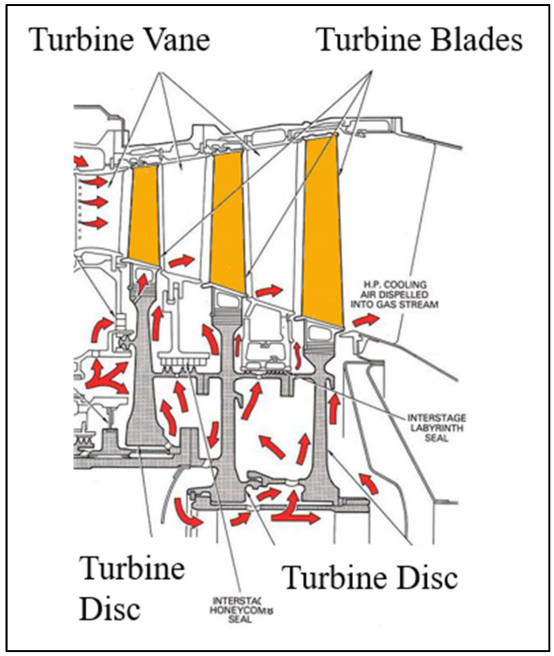


Figure 0-5 Schematic of a Turbine
modified from (Groh, 2013)

The rotor in turn consists of six components (shroud, airfoil, platform, fixing, disc and coverplate). This is shown in Figure 0-6

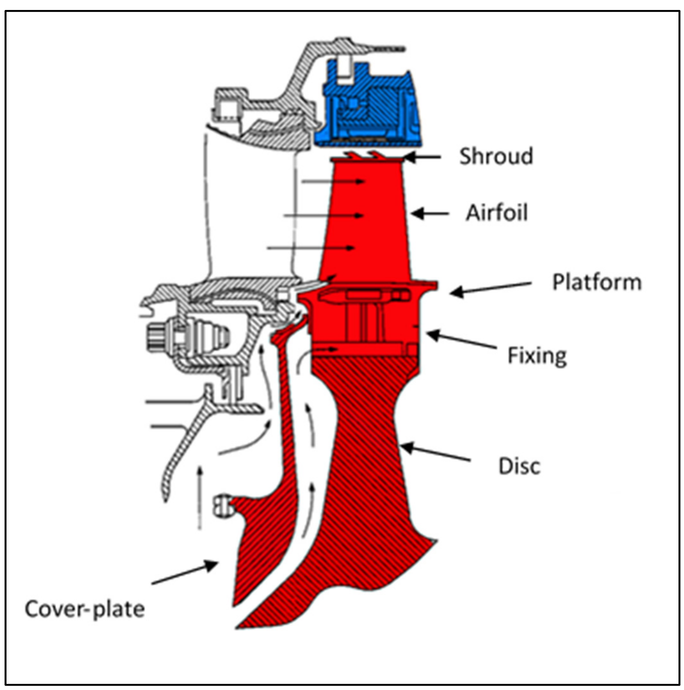


Figure 0-6 Representation of a Rotor

Since the turbines operate in temperatures above their melting point and in enormous centrifugal forces, they have more discipline interactions (aero, structural, cooling, air system etc. shown in Figure 0-7). This makes finding the optimal balance between all disciplines involved challenging.

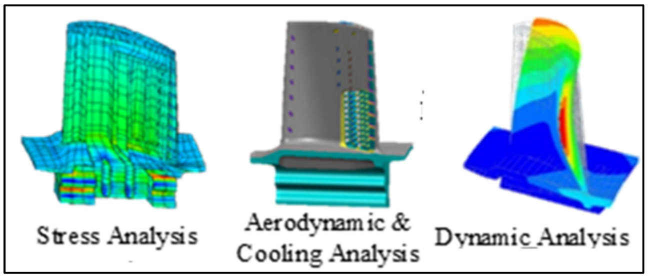


Figure 0-7 Disciplines Involved in Creating a Turbine Rotor

This point is further illustrated by Figure 0-8 that represents phase one of PDDS (focusing on the turbine). From this schematic, one can see all the different tools (Rotor Designer or iRSO, Tip Clearance Calculator, etc.) that would be created and the interactions between them. Figure 0-8 is not the artifact but rather an initial estimation of what tools and modules would be needed within PDDS.

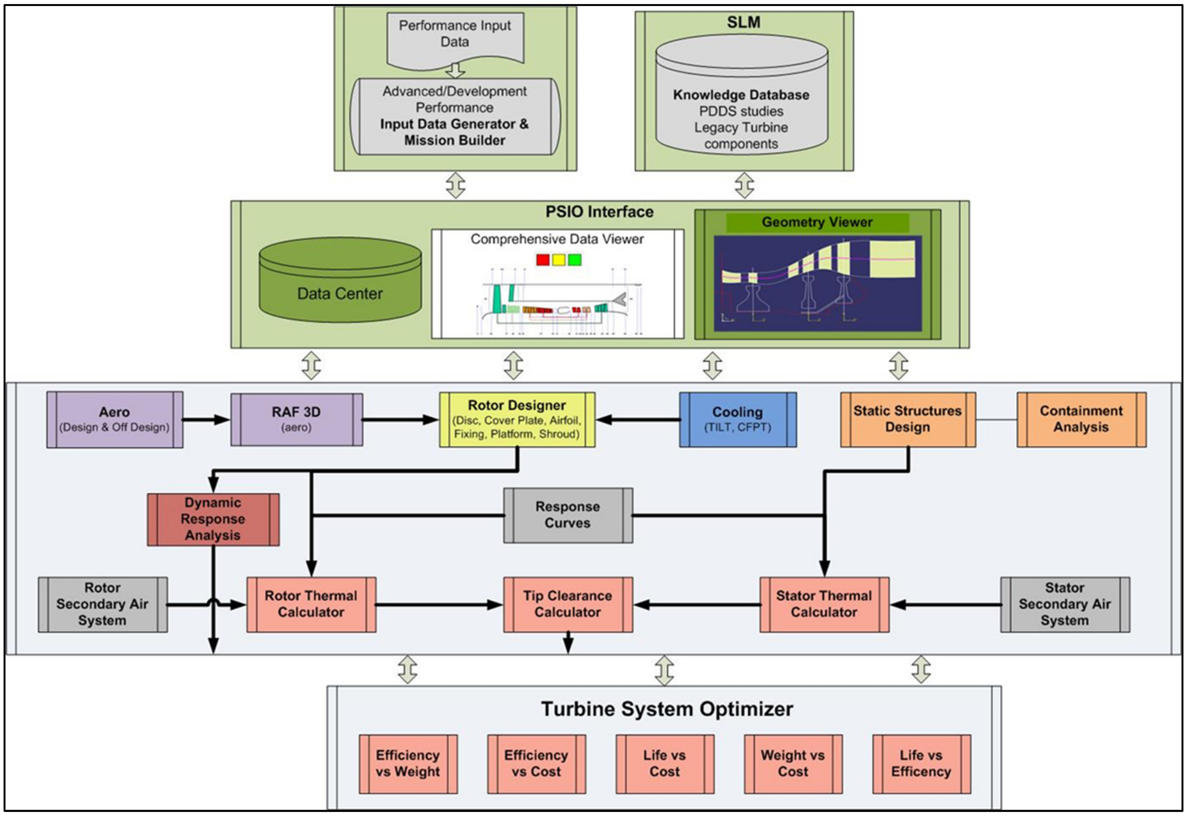


Figure 0-8 PDDS Schematic

From the work of Martins et.al. (2013) on MDO as well as Panchenko (2002) and NATO Science and Technology Organization (2006), it is evident that successful implementation of a PMDO (MDO at the pre-detailed phase) system would follow these following steps:

1. Develop a robust tool base – create tools that are able to work in a variety of situations and are able to evaluate various designs.

2. Apply single discipline optimization to individual analytical tools.

3. Create an integration framework – through a well-designed software architecture allowing for the tools to communicate seamlessly.

4. Implement multidisciplinary optimization with a clear statement of the design objectives, constraints and variables, and an appropriate selection of the algorithms.

The first three steps are the focus of this thesis as applied on the rotating components (airfoil, fixing, platform, disc and coverplate) of the turbine.

Wieringa (2009) describes “a practical problem as a difference between the way the world is experienced by stakeholders and the way they would like it to be”. To properly solve it, “the problem solver must identify stakeholders and their goals, derive design criteria from these, [and] investigate how a proposed solution would achieve a goal in a given context” (Wieringa, 2009). For the work presented, the design statement can be succinctly described as: creating an architecture for design optimization of turbine rotors during the pre-detailed design of gas turbine engines.

As mentioned before, the pre-detailed stage of the design of the turbines in a gas turbine engine is important as it is when the technical envelope of the engine is set. This phase of design according to the Research and Technology Organization (RTO) of NATO (2006) is however defined by tools that could be improved upon with respect to fidelity. Improvements to and automation of these tools could lead to better-tailored safety factors, and considerable improved life, operational and cost efficiencies. Tasks such as the copy-pasting and general conversion of an output of one program to an input of another can be avoided.

To mitigate uncertainty, this work aims to create an architecture for the turbine design and optimization as well as a larger architecture for future work on other engine assemblies (compressors, combustors etc.). The scope of this work is shown in the Bill of material (BOM) illustrated in Figure 0-9 by the items highlighted in red.

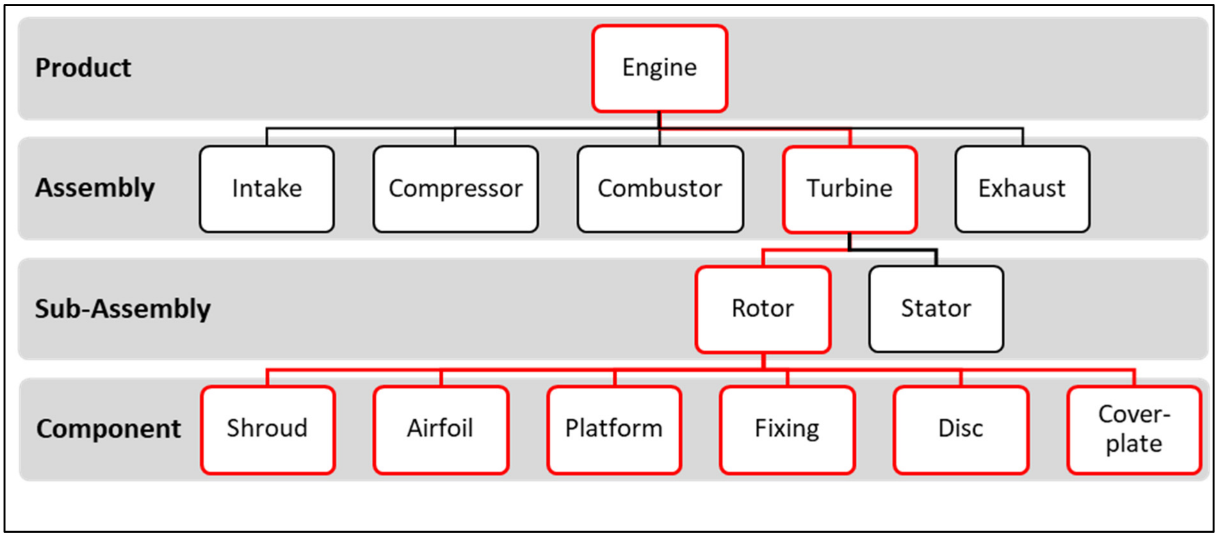


Figure 0-9 Engine BOM

This work creates a new architecture for the creation of multidisciplinary tools for the analysis of gas turbine engines. A new methodology to accomplish this research is devised in which power users are integrated into the research team, allowing them to be thoroughly engaged and gives the artifact a great advocate to the wider user base. A new methodology for the multi-disciplinary design and optimization of turbine rotors at the pre-detailed stage is also created in the implementation of the integrated rotor system optimizer (iRSO) module. It also creates a new module for the system level thermodynamic parameters (EDS) needed during the design of the turbine rotors. The adaptation of Figure 0-8 to focus more on the work presented in this thesis is shown in Figure 0-10. This figure also defines the three main terms that will be used to describe different parts of the final software artifact:

- System: Architecture for the development of all software
- Module: Compilation of tools for the analysis of a sub-assembly such as iRSO
- Tool: Piece of software that aids in the design and analysis of a component and is supported by one or multiple commercial software

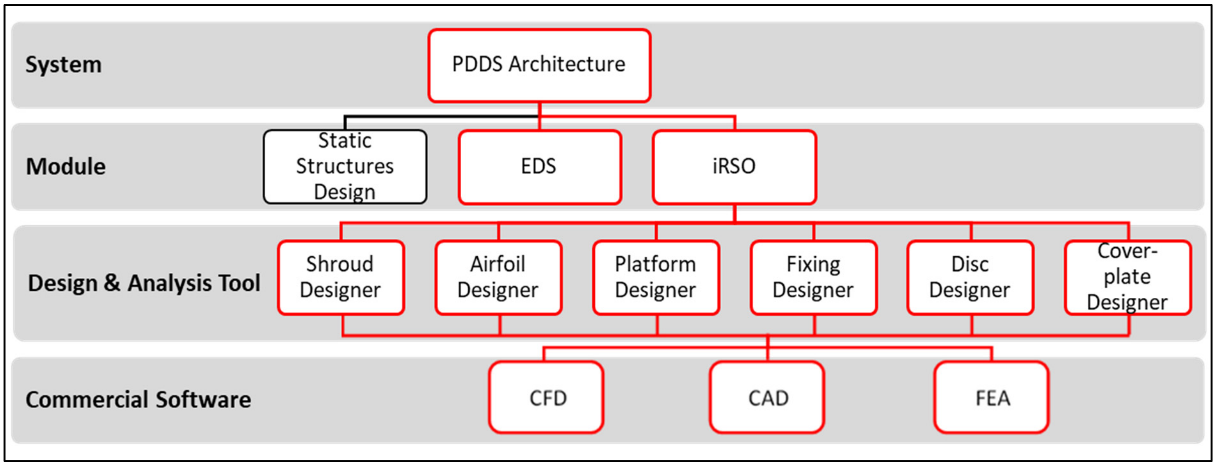


Figure 0-10 Hierarchy of Software Artifact Nomenclature

With such an ambitious project done as a collaboration between industry and academia, the selection of a general research methodological framework is critical; neither the design research method (DR) that ignores organizational dynamics, nor the Action Research (AR) method that only focuses on organization dynamics and not enough on theoretical rigour could be used. Throughout the creation of this artifact, Action Design Research (ADR) provided a methodological context for the solving of these practical problems as it leveraged the benefits of each of these two methods. “Business needs motivate the development of validated artifacts” (Wieringa, 2009) that meet the needs of the stakeholders and generalize the solution to solve similar set of problems (Sein, et al., 2011)

This thesis is organised according to the phases of ADR. CHAPTER 1 of this paper therefore focuses on the methodological framework, comparing DR to AR and the introducing the reader to ADR.

Chapter 2 describes the implementation of the first stage of ADR, problem formation, as well as providing an appropriate literature review.

Chapter 3 describes the BIE (building, intervention and evaluation) of the artifact explaining how artifact was created.

CHAPTER 4 explains the major BEI cycle learning from the first generation of the artifact and the revision to the objective and success criteria derived from this.

CHAPTER 5 is devoted to show the final artifact, providing examples of its use in a multi-disciplinary optimization of the Airfoil and the structural optimization of the Fixing. The impact on increasing the knowledge sharing in the early stage of rotor design and the benefits experienced are presented in the conclusion.

CHAPTER 6 highlights the learning that occurred throughout the creation and pinpoints the novel ideas introduced by this PhD.

It is also worth pointing out to the reader that this work is not about how optimization was done on any specific component but rather focuses on the architecture that was created to allow it to happen on all rotor components.

Due to the proprietary nature of this research and its potential to give Pratt and Whitney a large industrial advantage, some details have been withheld.

CHAPTER 1

METHODOLOGICAL FRAMEWORK

The goal of this chapter is to present different methodological frameworks and justify the choice of Action Design Research.

1.1 Presentation of three frameworks: DR, AR and ADR

Research into IT/S (information technology / information system), software and processes, like many engineering research projects, must satisfy a dual mission: it must make a theoretical contribution (and in doing so move the discipline forward) while at the same time solving a real-world practical problem. This is especially true when research is done in the industry as part of a collaboration between industry and academia as is the case with this work.

For a long time, a disconnect has existed between research and practical application. The goal and requirements mean different things to different people depending on what side of the street you are standing on. A misunderstanding usually ensues between the researchers and the users as to the goal of the final artifact (a tool, software, algorithm, technology etc.) of the research. The approach taken by the two sets of stakeholders are also at odds as the design oriented researchers and the organizational oriented users see different paths to the solution (Sein, et al., 2011). Usually, conflict arises as to how and how frequently to respond to the user changing requirements and the rigor that must be built into the artifact for academic needs (organizational researchers tend to care mostly that the artifact works perfectly to their specific needs and not to all the possible ways to the artifact could be used). The traditional approach to the research in IS, Design Research or DR, also does not recognize the role organizational dynamics play in the continuous shaping and reshaping of the requirements and the success criteria of the artifact during the development or even when deployed. Users must then find best ways to utilize the artifact and determine its commercial viability after the fact.

Using DR, the creation of the artifact is mostly separated into four distinct steps:

1. Requirements gathering – only step that the users are involved in.

2. Design – based on the requirements, decisions on what the tool is and how the tool will be made.
3. Development – the actual creation of the tool.
4. Testing – testing if the developed artifact meets the previously established requirements as gathered from step one.

The problem with DR is that the "understanding of the problem predates the development" (Sein, et al., 2011) of the artifact. This means that the success of the artifact is judged on how well it addresses the problems raised at the beginning and therefore missing opportunities to learn or uncover new problems throughout the development. The 'build then test' approach, as shown in Figure 1-1, also leads to potential rework or missed opportunities that earlier involvement of the user would have uncovered.

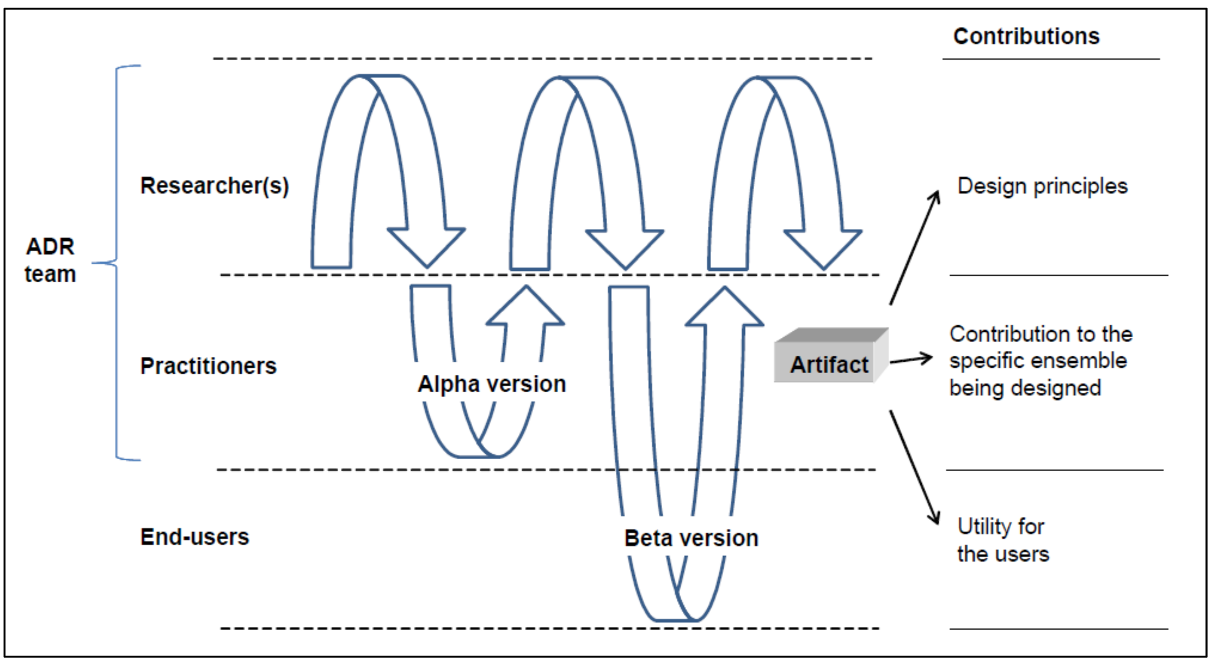


Figure 1-1 Design Research Methodology (Sein, et al., 2011)

Despite all the potential problem of this method of research, most people still use this gate method (definition -> proposed solution -> development -> test -> deployment to user) during research into IS because it is not without its positives. One of the main positives is that it is structured and regimented and since organizational intervention is rarely considered, the risk of mission creep (the increasing of requirements during development) is low. The methodology captures well all the intended consequences.

On the other hand, Action Research (AR) is somewhat the antithesis of DR. AR is best at solving immediate organizational problems. It's an iterative process where the requirements are not necessary fully defined at the beginning but are refined throughout the research as shown in Figure 1-2 through the multiple intermediary solutions being seen by the users. Research done using this methodology is iterative where the requirements are defined over "repeated cycle of inquiry" (Sein, et al., 2011). AR projects tend to be long and it is challenging knowing when you are finished or what success is defined as, but the result is an artifact that is tailored to the organizational needs with strong user buy in and an easily understandable place it plays in the overall day to day working. It also does a great job in capturing unintended goals.

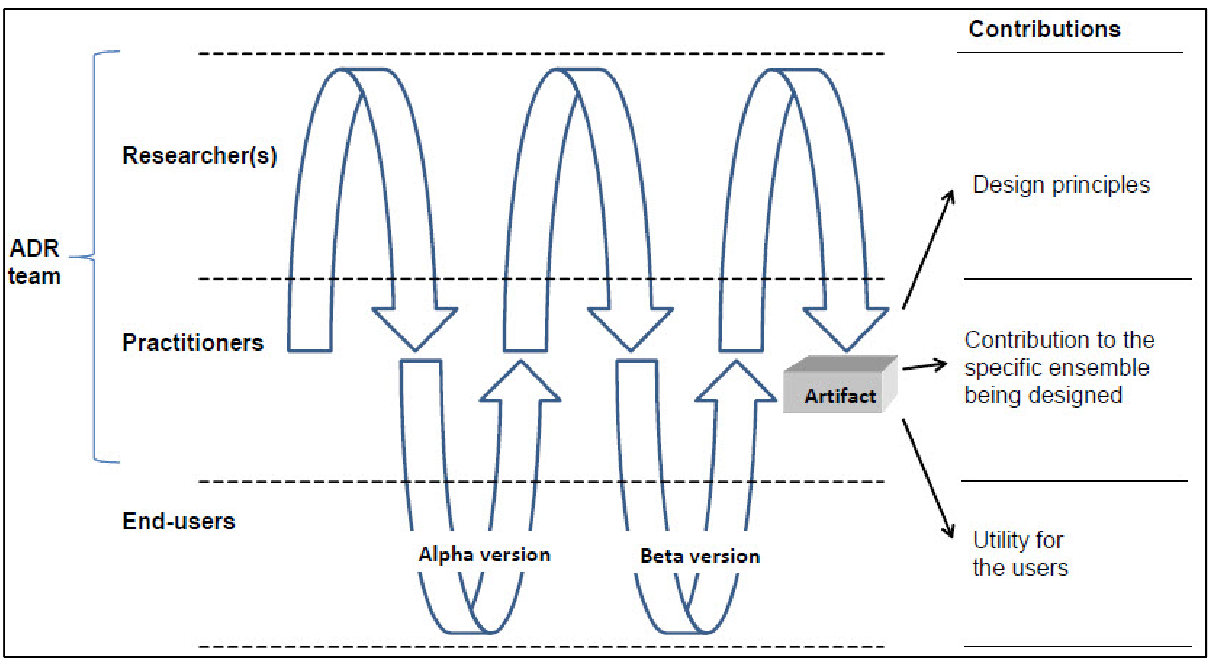


Figure 1-2 Action Research Methodology (Sein, et al., 2011)

For academic research done while embedded in industry both DR and AR approaches should work in tandem. Sein et. al (2011) call this amalgamation of AR and DR, ADR (Action Design Research). ADR seeks to capture both the intended and unintended requirements.

ADR is a research methodology for conducting design-oriented research in an organizational setting. It deals with an organization`s need to solve immediate organizational problems as well as the theoretical need of generalizing the solution as a large set of problems that could benefit from the insight gleamed in the solving the immediate one. ADR strictly denies the DR mantra of ‘design and then test’ in lieu of the AR approach of repeated cycle of requirements discovery through concurrent development and testing.

1.2 Stages and principals of ADR

ADR advocates for 4 stages in research (refer to Figure 1-3):

1. Problem Formulation.
2. Building, Intervention and Evaluation (BIE)
3. Reflection and Learning
4. Formalization of Learning

Stages 1 and 2 are in a cycle where problem formulation leads to BIE which leads to a better understanding of the problem which in turn leads to BIE and so forth. At any time, these two stages are supported by stage 3 where reflection and learning happens. When stages 1 and 2 have converged to an artifact that meets the intended and unintended requirements the artifact is complete and the learning can be formalized.

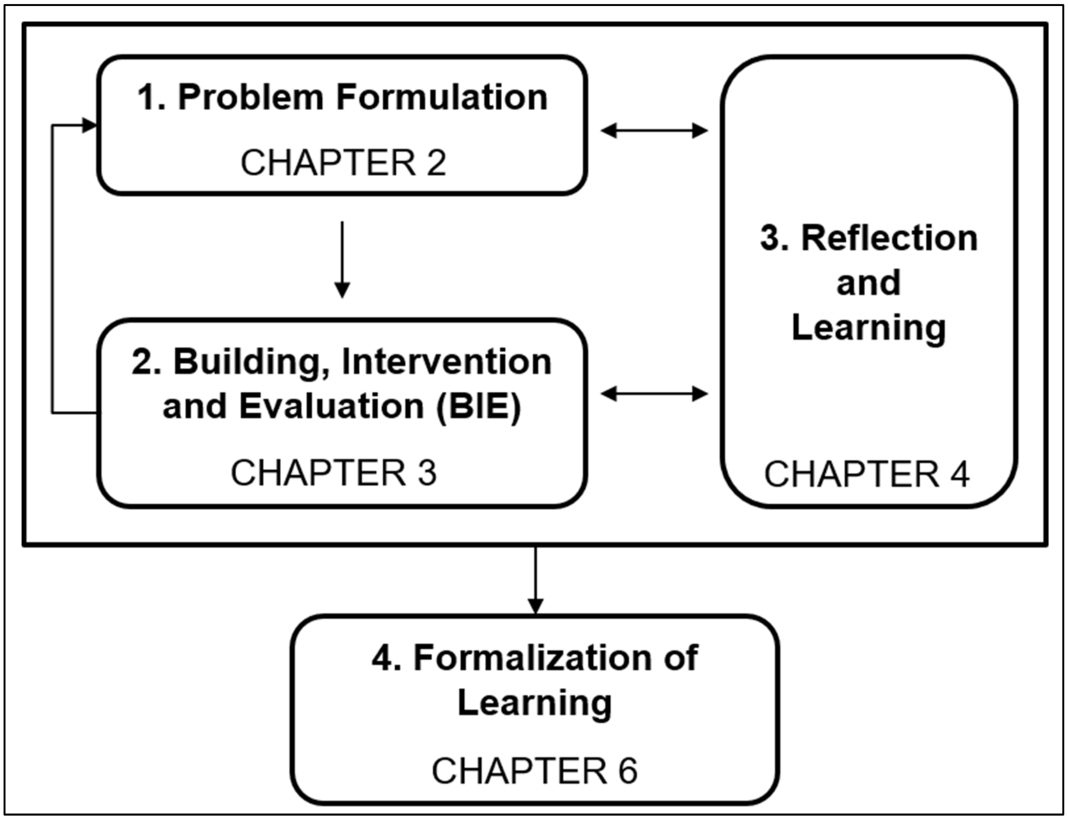


Figure 1-3 Stages of ADR Schematic (Sein, et al., 2011)

The first stage of ADR, problem formulation, deals with the understanding and description or anticipation of a problem either by the researchers or the users. In this stage the roles and scope of the research problem are defined and so are the roles that each member of the research team will play. The initial research question, which will be changed in subsequent cycles, is also defined at this stage. It is also imperative at this stage to secure long term funding for the project from the organization as well as commitment from the stakeholders within the organization. It is also worth having in mind that even at this early stage that the researchers should adhere to the ADR principle of "practicing inspired research" (Sein, et al., 2011). This means that even when defining the initial problem statements, it should be looked at as an example of a larger set of problems. Although the researchers are trying to solve the immediate organizational problem, the researchers should also make sure that proposed solutions are extensible to other problems that are similar to what is at hand. This is made easier if the solution is based on sound theoretical ground (another principle of ADR).

The second stage of ADR is concerned with the actual creation of the artifact. In this stage, as described by Sien et. al. (2011) one should endeavour to:

1. Discover initial knowledge creation target – understand what you are going to do
2. Select or customize BIE form – choose how the build, intervention and evaluate cycles will be done
3. Execute BIE cycle
4. Asses the need for additional cycles and repeat if necessary, through reflection and learning.

This stage of ADR relies heavily on three basic principles of ADR: reciprocal shaping, mutually influential roles and authentic concurrent evaluation. Through reciprocal shaping the "ADR team may use its chosen design constructs to shape its interpretation of the organizational environment, use the increasing understanding of the organizational environment to influence the selection of design constraints" (Sein, et al., 2011). It is also important to keep in mind that diversity in thought and experiences usually lead to better solutions. The researcher usually come in with sound theoretical knowledge but the users will have great organizational knowledge and understanding of the practicality of the solutions. Learning from each other and letting each other affect each other's domains most often than not leads to a better solution; in other words, the artifact should shape the organization while being shaped by it, different members of the research team should learn from each other and the team should constantly be evaluating what it makes.

The third stage is where reflection and learning occurs and is hinged on the ADR principle of guided emergence. The juxtaposition of the guided emergence might seem oxymoronic, but it reflects the thought behind the principle. The artifact should not only reflect the initial requirement (guided) but also allow for the emergence of new criteria.

Finally, the last stage is the formalization of learning. The problem must then be classified as a larger set of problems where the solution can be generalized to solve this large set of problems. This thesis is one attempt at formalizing the knowledge created during the research.

The thesis presents the work within the research framework of ADR. Each chapter is dedicated to a certain stage of AD, shown in Figure 1-3 with the presentation of the final artifact presented in CHAPTER 5. The next chapter is dedicated to the problem formulation stage of the research.

CHAPTER 2

PROBLEM FORMULATION AND LITERATURE REVIEW

This chapter sets out to define the thesis and design goal, justify technological decisions and finally define the artifact's requirements.

2.1 Thesis and Design goal

As described in the Introduction, correcting a suboptimal decision made early in the design process is very difficult and often expensive as the ability to influence change is greatly reduced over time (Figure 0-3). Through this research, the hope is that the artifact created would allow for the knowledge to be brought forward in the design process to the pre-detailed phase where more design freedom is available. As part of a P&WC program on improving the preliminary design system with the goal to deploy a fully automated and integrated propulsion system optimizer, this research is focused on creating a system level architecture (refer to Figure 0-10) that enables the multi-disciplinary optimization of a turbine rotor.

Because this thesis so heavily leans on the design of this artifact, the goal of the thesis and the design goal are one and the same. As mentioned by Wieringa (2009), a proposed solution for a practical problem would have design criteria derived from stakeholder goals. Given that this research occurs in the framework of a larger project with the goal of deploying a fully automated and integrated propulsion system optimizer (see INTRODUCTION), the goal of this specific researched would need to aid in the achievement of this overall goal. For this research the goal is to:

> *Deploy a fully automated and integrated turbine system capable of optimization that would allow for knowledge to be brought forward in the design process to the pre-detailed phase where more design freedom is available enabling PWC to conduct engine development faster and reduce risk when it comes to responding to RFP from airframe OEMs.*

In other words, devise an artifact that allows designers to explore more options with greater confidence in the pre-detailed phase by obtaining answers to “what if” questions.

This leads to the creation of an integrated rotor system optimizer (iRSO) module and an engine data system (EDS) module (this system, modules and tools that exist will be referred to holistically as the software artifact).

The thesis counts two objectives:

1. Design and build the artifact (CHAPTER 3, CHAPTER 4)
2. Test and evaluate the artifact (CHAPTER 5, CHAPTER 6, CONCLUSION)

The final artifact will be shaped by the numerous technological choices to be made. To be able to define the objectives and requirements of the artifact, a few basic questions need to be answered as suggested by the NATO RTO Research Task Group (NATO RTO Research Task Group AVT 093, 2006):

- Which overall design approach should be taken? The answer to this question greatly influences the architecture.
- How should geometry be represented?
- How should the component design be approached?
- How can the problem complexity be approximated?
- Which optimisation strategy should be deployed?

The literature review conducted in the following sections focuses on specific technological decisions that needed to be made to answer the questions above so as to facilitate the creation of the software artifact.

It is worth noting that most theses present a review of current work. However, due to the fact most gas turbine engine manufactures do not have a large number of engine families, like P&WC does, most engine manufactures may not view this research as cost effective enough to undertake due to the low number of engine concepts they entertain per year. Those that might have conducted such research do not publish their findings due the potential of this research to yield a large commercial advantage, therefore, the traditional literature review is near impossible to accomplish.

2.2 Technological Choices and Literature

In an optimization problem especially similar to the one the author is proposing to embark on, many different aspects hold the key to success. Decisions made on the modelling used to represent geometry, design methodology used, approximation techniques utilized (if any) as well the optimization algorithm itself are crucial. A review of previous works on these topics are presented in this chapter.

2.2.1 Multidisciplinary Analysis and Design Environment Approaches

As stated in the NATO technical report on integration of tools and processes for affordable vehicles (NATO RTO Research Task Group AVT 093, 2006), two different paradigms have existed in multi-disciplinary analysis and design (MAD): “Monolithic Approach (MA)” and “Best in Class Approach (BCA)” systems. MA systems consist of a single tool that has all the “functionality or tools to model, analyze, and optimize a given component or entire configuration” (NATO RTO Research Task Group AVT 093, 2006). MA tools are easier to maintain and are robust but are often lag behind BCA systems because they do not contain the latest design and analytical tools for all disciplines that BCA systems have.

BCA systems on the other hand have the best tools for each disciplines, and typically use a “scripting language to ‘glue’ together several independent applications” (NATO RTO Research Task Group AVT 093, 2006) to provide the best design and analysis methodology for each discipline. A BCA system might have a CFD software such as CFX or Fluent to deal with the aerodynamic questions and a FEA specific software such as ANSYS or NASTRAN and a CAD such as CATIA or SOLIDWORKS all “glued” together by a program that controls when to call each of these disciplines specific software when needed.

In general, MA systems are “usually built around a single discipline expertise such as mechanical analysis, CAD, and have marginal capabilities in other disciplines such as controls, or computational fluid dynamics" (NATO RTO Research Task Group AVT 093, 2006). This makes the utility derived from these systems not universal as some engineering disciplines do not get the necessary fidelity for their respective discipline. On the other hand, BCA systems give the users access to the "best" technology available for a given discipline. At P&WC, different departments and disciplines choose between MA and BCA as they see fit. The groups that are responsible for the integration of the final product (such as Turbine Design) tend to favour MA while other disciplines such as Structures and Aerodynamics tend to prefer BCA as it gives them access to the "best" available technology. Given this fact, BCA is therefore the best approach for this research work as engineering experts can use the software that they are more comfortable with. It does however give large importance to data structure as each of these individual discipline specific programs require data in its on format.

2.2.2 Geometry Modelling

The optimization of a turbine rotor, like any other geometry optimization, is crucially dependent on a good modelling of the geometry. The modelling may be in the form of mathematical equations, a set of curves or a parametric model; the one common thread is that the model must be "an efficient and flexible description of the geometry" (Sobieszczanski-Sobieski & Haftka, 1997); simply put, a model needs to be able to describe as many different types of geometries for a given component (airfoil, platform, fixing, disc and coverplate as shown in Figure 0-6) in the least amount of parameters as possible. Each of these components are important in the final rotor and have different difficulties when it comes to modelling them.

Extensive work has been done in the modelling of turbine airfoils with many authors using sets of curves. One such example is the use of Bezier curves in the work of Goel (2009) and Tayla et.al. (2002). In some works (Saric, et al., 1996), a geometry based on company private parametric models have been used.

Due to the familiarity of Computer Aided Design (CAD) software in the aerospace industry, in this work, parametric models created in CAD software are utilized to define the geometry.

These models were the work of previous researchers [(Guélin, 2014), (Ouellet, et al., 2014)] who worked under the PDDS ETS-PWC Research Chair and have been proved and validated robust and representative of real geometry. Ouellet et al. were able to create a parameterized axi-symmetrical disc that could accurately represent most discs while Guélin was able to model the coverplate. More details of the geometrical models created are presented in sub-chapter 5.3.

2.2.3 Direct vs Inverse Design Method

Many methodologies exist in the design of turbine rotor; among them are the direct design and the inverse design methods. Both these methods afford a designer control on some aspect of the design process while inadvertently restricting control in another aspect.

The direct design method, in which the designer changes the geometry iteratively until desired performance is achieved, affords direct control of the design parameters (as described in paragraph 2.2.2). For the design and analysis of rotors, it would be described as:

1. Creation of initial geometry - this can be either an existing turbine rotor that is being re-designed or a first guess of what the geometry should be
2. Analysis of the rotor through analytical tools such as CFD or FEA – this can be a single discipline analysis such as Aerodynamics or the multi-disciplinary analysis where the rotor is analyzed for stress, aerodynamics and thermodynamics
3. Assessing the success of design by comparing results from a set of performance or structural objectives
4. Alter the geometry to affect positive change
5. Repeat Steps 2 – 4 until the set of performance and/or structural objectives are met.

This method is preferred by engineers because it is intuitive as it lets the designer incrementally alter the geometry to predict the flow or structural integrity. It has however been described as "very laborious" (Jha, 1999) and inefficient (Dulikravich, 1991). The method also "requires considerable insight" (Jha, 1999) as the designer also "requires some a priori knowledge of design sensitivities" (Tayla, et al., 2000) to be able to alter the geometry in the correct way to affect positive change in the analysis results.

Inverse design methodology affords the designer the ability to predict the geometry based on a known flow field or stress distribution. When the flow field or stress distribution is known, this method is considered more efficient. This methodology however offers "little control over blade geometry" (Goel, 2009) and "it does not adapt as readily to the application of geometry constraints as direct methods" (Tayla, et al., 2002). Consequently, the geometry predicted may be "mechanically infeasible" (Goel, 2009), i.e., cannot be fabricated with the current manufacturing technology.

Another drawback of the inverse design methodology is that it creates "strictly point-designs rather than ranges" (Dulikravich, 1991), the design performance is only achieved at design point and might not perform favorably if the operating condition changes. A possible solution to this problem however is to choose inputs (flow and stress distribution) that are broad enough to fulfill all operating conditions.

For this work, the Direct Design Method shall be used due to the organization needs and familiarity. P&WC has used this method of design as it is seen to be more intuitive, allowing the engineer to affect change as they would like. Insight shall be injected to the process by maintaining a database of successful design and using them as references.

2.2.4 Approximation Techniques

During a single optimization of a turbine rotor, the numerical analysis code may be called on dozens, if not hundreds, of time. The use of a full definition 3D or even quasi 3D solver all the time may be prohibitive due to the high computational cost and time associated with the multiple runs. It therefore necessitates the deployment of easy-to-use, fast approximations in the running of the initial optimization loops.

Many techniques exist that assist in the approximation of complex systems but few that are able to properly approximate non-linear, highly coupled, hierarchal problems such as the optimization of a turbine rotor geometry to minimize weight, while increasing performance and rotor durability One such method is the Global reduced – basis approximation.

In the Global reduced –basis approximation, the results are estimated by "linear Taylor series and the reciprocal Taylor series" (Jha, 1999). An example of this is shown in the work of Longo et.al. (1991) where the authors where able to approximate the structural responses by a Taylor series in the reciprocals of all or some of the variable.

The Variable-Complexity Modelling (VCM) is another means to reduce the computational cost associated with optimization. As shown by Hutchison et.al. (1994) in the design of high-speed civil transport, VCM is a process in which multiple levels of complexity are used in analysis, with the analysis of each level of complexity calibrated with the analysis from a higher level of complexity. An example would be the use of simple '1-D' stress analysis that only looks at variation in the stress in one cardinal direction and is then calibrated by the 2-D results (results from the '1-D' are multiplied by a factor equaling previous 2D/1D results) which in turn are then calibrated by 3-D results. This method of approximation lends itself very well to this research as it provides upon request fidelity; the use of high-fidelity techniques (2D or 3D) when required and low (1D) when not. It is also preferred because it allows the research to take advantage of both the high and lower fidelity tools that P&WC has, another example where the organization shapes the research decisions.

2.2.5 Optimization Algorithms

The optimization of a turbine rotor is very complex and includes many coupled constraints. These constraints often lead to conflicting requirement. A good example is the fact that higher efficiency in a rotor is obtained with a higher turbine inlet temperature which leads to rotor structural integrity concerns as the higher temperature increase the possibility of creep failure. To counteract this, novel materials with better creep resisting properties may be deployed (which costs more money) or a cooling scheme needs to be introduced. Since cooling air is taken after the compressor stages, it leads to lower overall engine efficiencies and leads to a heavier rotor as it needs to be larger to accommodate the cooling holes. The optimization algorithms need to be able to address these multiple conflicting requirements.

One of the simplest optimization algorithms that engineers have relied upon is the gradient derivative approach. This optimization approach looks at the previous and current value and evaluates the gradient between the two. If the gradient is in the intended direction (i.e., it is negative for a minimizing objective or positive for a maximizing objective) then the geometry changes are continued in that direction if not then the changes are reversed.

This approach works well for simple optimization problems that have single minima or maxima but can be problematic with more complex problems as it has the susceptibility to get stuck on a local extremum and never find the global extremum. This can easily be demonstrated when one looks at a graph of a function with both local and global extremum such as graph of a simple septic function as in Figure 2-1. In this case gradient derivative approach, depending on the initial guess (labels 1-6 in Figure 2-1), the optimized minimum would be different. Guess 1 and 2 will get stuck at point A while 3 and 4 will get stuck at point B, only guess 5 and 6 will achieve the true minimum at point C.

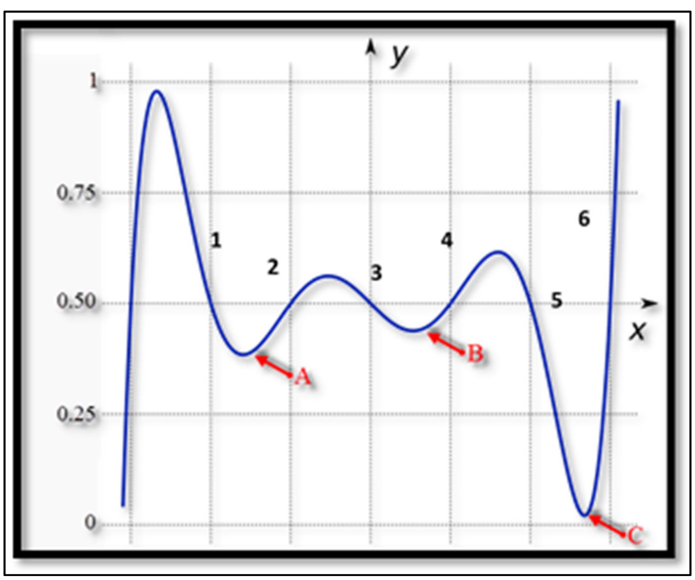


Figure 2-1 Simple Septic Function

Global Sensitivity Equations (GSE) could also be used with great effect in optimization problems. "System derivatives are computed from a set of linear algebraic equations, the GSE's, whose coefficients are the partial derivatives of outputs with respect to inputs" (Sobieszczanski-Sobieski & Haftka, 1997). This method establishes the dependencies of each design variable to the outputs and then optimizes the equation to the objectives. Although this method seems ideal as it would offer a deep insight into how each design variable affects each output, the sheer scale and coupling of variables used in the design of turbine rotors makes it impractical as it would require the constant use of super computers or would take an enormous amount of time on regular computers.

A more practical way to utilize sensitivity analysis is through Concurrent Subspace Optimization (CSSO). This optimization algorithm separates the variables that fall into a disciple (or subspace), creating a subset of the design variable and if a variable affects more than one discipline, it is included in all the subsets it affects. These subsets are used in the concurrent optimization of each subspace. Optimized subspace solutions then become the target.

CSSO has the possibility of being extremely efficient but it depends on the scarcity of the system (i.e., how many variables exists in multiple subspaces). In extreme cases where all design variables affect every constraint directly i.e., each subspace includes all design variable, the benefit of CSSO is greatly reduces and is not as desirable. This would be the case in the optimization of turbines as most variables will affect multiple discipline results and some variable affecting all disciplines.

Collaborative Optimization (CO) takes the decomposition of the optimization problem further than CSSO. It eliminates the need for separate system and the sensitivity analysis optimization at the end by "blending the design variables and those state variables that couple the subspaces (variable that define the interactions between disciplines) in one vector of the system level variables" (Sobieszczanski-Sobieski & Haftka, 1997). The difference between CO and CSSO is that the subset of design variables need not be unique to each subspace and that they may exist in multiple. Both the "system state variable and design variables are manipulated by an optimizer until the solution and the optimum are both attained" (Sobieszczanski-Sobieski &

Haftka, 1997) and hence is considered simultaneous analysis and design (SAND). Very much like CSSO, CO's effectiveness is greatly dependent on the scarcity of the problem and like CSSO, because of the coupling in the design of a turbine rotor, this method might not be feasible.

Decision Support Problem (DSP) combines mathematical rigor with concepts from goal programming to find variables that satisfy constraints and achieve conflicting goals. "It provides a means for modeling decision encountered in design, manufacture and maintenance" (Mistree, Hughes, & Bras, 1992). DSP also provides support for human judgment in selecting of preferences based on multiple attributes among several alternatives, the compromise between different solutions, decisions that involve hierarchical systems and risk mitigation. Since not all requirements are equally important for an engineer, DSP allows for the assignment of weight to each requirement.

It is important to emphasize, it is not a goal of this research to build an optimizer but rather to create an artifact that is optimization capable. To illustrate the created software artifact's feasibility to conduct optimization, an in-house optimization algorithm, described in section 5.6.3, was used. The algorithm uses a surrogate model that it builds and updates through each iterative runs to predict best results using the gradient method and then runs what the surrogate model predicts to be the candidates with the best results. This surrogate assisted optimization (SAO) algorithm also allows for weighted functions as in the DSP method. The organizational familiarity with this in-house optimization algorithm made it the best choice.

2.3 Objectives

From the literature review (NATO RTO Research Task Group AVT 093, 2006) it is apparent that a PMDO implementation must have the four technical objectives. The following is a reproduction of the technical objectives:

1. Product Representation — A way to represent the product along with the design intent/rules.
2. Seamless access to varying fidelity best in class tools to evaluate/modify the design.

3. Process Representation with Secure Communication between all tools, data, and vested parties involved in the product development process.
4. Modularity that enables high level of reuse when moving from one application to the other.

Coupled with these technical objectives, a fifth objective coming from the business aspect of the PDDS project was to:

5. Assist in getting and keeping key stake holder buy-in and secure long-term funding

Looking at both the technical objectives and the business objective, a few things become evident. With this research being done in industry where the organizational needs are paramount, demonstrating value of the artifact as fast as possible in the project will be key. The artifact should also reduce the time taken for engine development by a factor of 10 (this factor was an initial goal of the business) and the amount of tool specific training needed before one could design a component.

The technical objectives 2 outlined by NATO (NATO RTO Research Task Group AVT 093, 2006) would aid in the achievement of the artifact's objective of reducing engine development time as varying level of fidelity analysis will aid in the speeding up of the engine development time. Lower fidelity tools could be used when speed is essential and high-fidelity tools when necessary. Technical objective 4 will aid in the quick delivery of useful software as required by P&WC while lowering the experience barrier to design successful components will provide further value to the P&WC.

Technical objectives 3 as outlined by NATO (NATO RTO Research Task Group AVT 093, 2006) is mirrored in the artifact objective 4 where engineers are kept in control of the design process, while technical objective 1 is mirrored by artifact objective 5, ensuring that all components and sub-assemblies are able to modelled with sufficient granularity.

For the first generation of the artifact, the objectives were:

1. continuously provide value to P&WC.

2. reduce the design time of turbines by a factor of 10.
3. reduce tools specific training needed to design a component; The training referred to here is training required to be a user of the tool and does not refer to the decade or so it normally takes an engineer to become skilled enough to understand the technical implications of what this tool will output as data.
4. facilitate engineers to control the design process and to inject judgement and creativity.
5. have flexibility in modeling and analysis that allows the reproduction of all current rotors design and creation of future design.

It is worth highlighting to the reader that since this research was conducted using the ADR methodological framework (which advocates for iterative BIE cycles), these objectives were further refined and appended to as new objectives emerged through reflection and learning as will be seen in CHAPTER 4.

In lieu of the traditional literature review, for the reasons specified in the beginning of CHAPTER 2, the literature review done aided in both refining objectives, requirements and the design/technical choices that would need to be taken to achieve those requirements. Table 2-1 outlines the initial requirements and the design/technical choices taken to achieve them.

Table 2-1 Initial (Generation 1) Objectives and Requirement Matrix for Research

Gen.	OBJECTIVE	USER REQUIREMENTS The system shall:	DESIGN / TECHNOLOGICAL CHOICES
1	1. Continuously provide value to P&WC. 2. Reduce the design time of turbines by a factor of 10.	1. allow for the rapid execution of analysis - reduce the time taken to create and analyze rotors by 10X during the pre-detailed design.	1. Seamless Access to varying level of fidelity analyses to ensure that the right tool is used for the right job while minimize the time taken.
		2. be scalable to allow an increase in the number of users as the system and tools mature.	
		3. allow inexperienced software designers and programmers to develop good software as the research team will have varying degree of experience with coding (some with none) but quality needs to be maintained.	1. Standardization of coding practices.
		4. be rapidly deployed to ensure customer satisfaction by early and continuously delivering valuable software.	1. Familiar software and analytical methodology shall be used. 2. Use aspects of Agile programming.

Table 2-1 Initial (Generation 1) Objectives and Requirement Matrix for Research (Continued)

Gen.	**OBJECTIVE**	**USER REQUIREMENTS** **The system shall:**	**DESIGN / TECHNOLOGICAL CHOICES**
			2.1. Close, daily cooperation between software developers and client. 2.2. Co-location to ensure frequent face-to-face conversations. 2.3. Simplicity in the code. 3. Use aspects of 'pure' waterfall model of distinct phase of coding and testing. 4. Tools developed independently; integration shall occur in future versions.

Table 2-1 Initial (Generation 1) Objectives and Requirement Matrix for Research (Continued)

Gen.	OBJECTIVE	USER REQUIREMENTS The system shall:	DESIGN / TECHNOLOGICAL CHOICES
	3. Reduce tools specific training needed to design a component. 4. Facilitate engineers to control the design process and to inject judgement and creativity.	5. be easy to use. 5.1. shall require minimal tool specific training before one can effectively use the tool.	1. Shall have a single graphical interface that will display all inputs, CAD representation and analytical outputs independent of the tools used to generate the data.
	5. Have flexibility in modeling and analysis that allows the reproduction of rotor designs and creation of future designs.	6. Facilitate geometric representation of components.	1. Using commercial CAD software, model all components and features.

CHAPTER 3

BUILD, INTERVENTION, EVALUATION (BIE)

The second stage of ADR as previously mentioned deals with the actual creation and evaluation of the artifact. As is customary, this stage has multiple feedback loops with the problem definition stage as the reciprocal shaping of the artifact and organizational dynamics, through constant evaluation, tend to redefine requirements. One of the first things was to agree between the research team and P&WC how the BIE loops were going to occur; whether the BIE loops were going to be like an IT/S dominant BIE, organizational-dominant BIE or somewhere in between the spectrum. With iRSO being a CAD heavy module, a good understanding of CAD parameterization is needed to create flexible, robust parameterized model. iRSO's need to have multiple links with outside commercial and in-house software and the fact that the team had varying degrees of comfort in these programs meant that a clear coding methodology and framework was needed to assure that work that was done by one member of the team could seamlessly plug into the larger picture. Finally, iRSO needed to allow reciprocal shaping and authentic and concurrent evaluations to reap the greatest rewards from the tool. This chapter documents the BIE process used, as well as giving multiple examples when adhering to the principles of ADR helped along the way.

3.1 BIE Schematic

The BIE process chosen for the development of the software artifact was neither the generic IT/S dominant one, see Figure 1-1, (Sein, et al., 2011) nor the organizational dominant, see Figure 1-2, (Sein, et al., 2011) one but rather took the best from both. As shown in Figure 3-1, the BIE process introduced a limited user base (or power users) that neither IT/S dominant nor organizational dominant BIE would have. These power users were not a mere subset of the user community as would be the case in normal Beta version release of an artifact in IT/S dominant processes but were experts on what the software artifact was going to replace and in some cases were heavily involved in the definition of requirements. This set of power users were critical in the evaluation and intervention of the software. The power users were given an advance peek of the software artifact as often as possible (alpha versions and beta version). An alpha version is the tool with a new feature that has not been completely validated and a beta version is the tool with a validated new feature but not all the features outlined in the requirements. The power users would test these initial versions before a beta version would go to the larger user community. The integration of power users allows them to be thoroughly engaged and gives the artifact a great advocate to the wider user base.

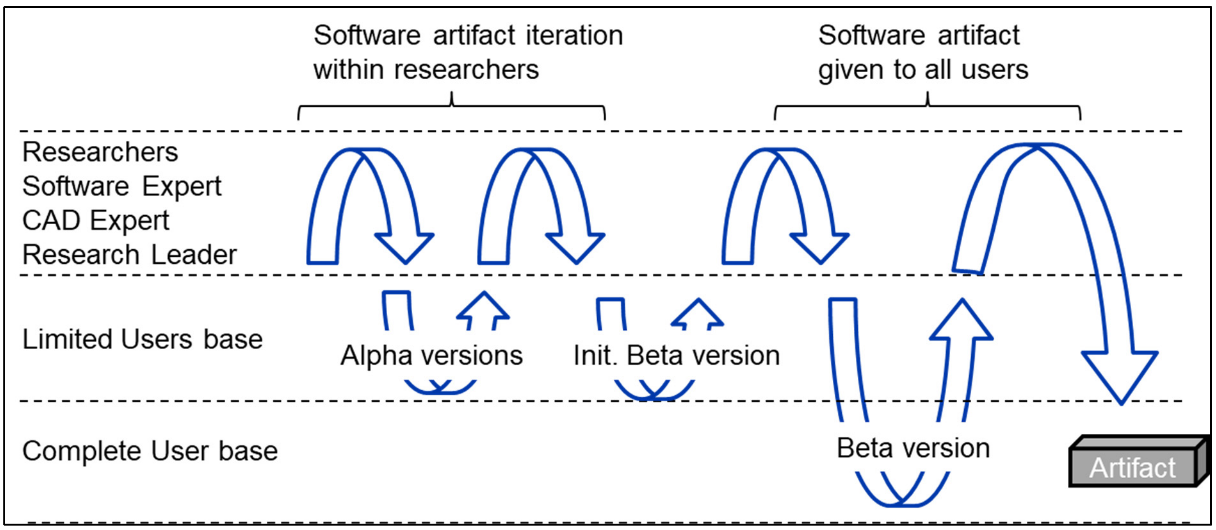


Figure 3-1 iRSO BIE schematic

Two major BEI cycles, generation 1 and generation 2 of the artifact, were executed (along with numerous minor cycles). The resulting artifact of each cycle was given to the power users to evaluate if it met the requirements set forth at the beginning of that cycle. Through the reflection and learning (step 3 of the ADR method (see Figure 1-3), the problem was reformulated (step 1 of ADR, Figure 1-3) and requirements refined. Some of these cycles helped the research team learn of the best CAD parameterization to be used and some resulted in new learning on how to best create an integrated set of tools. The lessons learned from the two major cycles and the influence the first cycle had on the requirements is presented in CHAPTER 4. CHAPTER 5 only deals with the final artifact.

3.2 ADR Principal 3: Reciprocal Shaping

Traditional approach to IT/S research doesn't recognize the role that organizational dynamics plays in the continuous shaping and reshaping of the requirements and success criteria of the artifact in development or when deployed. When the artifact is deployed, users need to find how best to work with the artifact's strengths and limitations instead of the artifact fitting with the environment like a hand to glove. In most cases, this is understandable as big software artifacts are usually used in multiple organizations, with each organization having different dynamics, customizing to each is an expensive process. Even when this is not the case and an artifact is built for one organization, the users are kept at arm's length as they are seen to cause delays; the client tells you what they want, and you build it for them. The reciprocal shaping principle of ADR changes this.

Reciprocal shaping "emphasizes the inseparable influences mutually exerted by the two domains: the IT/S artifact and the organizational context" (Sein, et al., 2011). It allows the creation of an artifact that not only allows people to work how they would like but also through its creation changes aspects of how they work. Throughout the work conducted, this principal was strictly adhered to and led to great results. An example that clearly illustrates this is in the setting of the requirements and definition of success.

CHAPTER 4

REFLECTION AND LEARNING

4.1 Evolution of Success Criteria

At the inception of the artifact, focus was mainly on the execution time reduction in the pre-detailed design phase. The goal was to reduce the time it took to do what they already did by at least a factor of 10. This was an ambitious goal but not exactly out of the realms of possibility. Since this artifact is a pre-detailed tool, the factor of 10 reduction was to be applied to only the time it took for an engine to go through pre-detailed design.

The team endeavoured to accomplish this goal but at one of the first bench marking we were not exactly meeting that goal; the time had been reduced by a significant amount but alas not 10-fold. At that point, an in-depth analysis of what the iRSO module did and offered was conducted and a few things were discovered:

- iRSO offered a significant higher fidelity of analysis to what was available before at the pre-detailed design stage.
- In some cases, iRSO allowed the analysing of some aspects of the design that were until then the strict purview of the more fidelity focused detailed design.
- iRSO enabled a fully 3D Rotor to proceed into detailed design at a more defined level meaning that the detailed design stage could be shorter.

These discoveries led to the realization that although iRSO wasn't exactly reducing pre-detailed design by a factor of 10 but it was reducing the much more expensive detailed design phase by a cycle or two. iRSO also allowed P&WC to lower the risk of the designs leaving concept design phase and pre-detailed design phase.

Understanding all of this, the objective and success criteria changed to:

- Creating a system that is scalable for future expansion.
- Lowering the risk of the designs leaving concept design phase and pre-detailed design phase.

The definition of some requirements evolved such as the reduction of time, that was limited to just the pre-detailed design was changed for gen. 2 to include detailed design. This means that even if the pre-detailed design phase time isn't reduced by 10X this requirement could be met through a reduction in detailed design time. This change in requirements, would still allow the objective of reducing risk in responses to bids to still be accomplished. Hence the artifact allowed the organization dynamics to change and the artifact focus to change, the artifact changed P&WC and P&WC changed it in turn.

The objectives, requirements and the design/technology choices for the new generation of the artifact is presented in Table 4-1. All the objectives outlined for the generation 2 of the artifact are in addition to those outlined in Table 2-1 and in most cases so are the requirements (apart from req. 1 and req. 7). As mentioned earlier, req. 1 replaced req. 1 from generation 1 (seen in Table 2-1). The lessons learned from the first generation of the artifact also led to the refinement of requirement 6 from the generation 1 of the artifact (seen in Table 2-1). The artifact would now only model those features that would affect the architecture of the turbine and not every single feature. This would speed up the artifact's execution time much like access to multiple levels of fidelity would decrease the time taken for analysis by using only the definition/ analysis needed to make an informed decision. This is shown in req. 7 in Table 4-1.

For the sake of readability, only the new or redefined objectives / requirements that were introduced in the second generation of the artifact are show in Table 4-1. Apart from req. 1 (which was redefined as req.1 in Table 4-1) and req. 6 (which was redefined as req.7 in Table 4-1) of generation 1, all other requirements (req. 2-5) still applied to the second generation of the artifact.

Table 4-1 Revised Objectives and Requirements Matrix for 2nd Generation of the Software Artifact

Gen.	OBJECTIVE	USER REQUIREMENTS The system shall:	DESIGN / TECHNOLOGICAL CHOICES
2	1. Create a system that is scalable for future expansion. 2. Lower the risk of the designs leaving concept design phase and pre-detailed design phase.	1. reduce the time taken to create and analyze rotors during the design (both pre-detailed and detailed design).	
		2. be extensible. 2.1. to allow growth of data structure so that future expansion of the data structure to include other assemblies of the engine (compressor, combustor etc.) that would incorporate other disciplines. 2.2. to allow the addition of future functionality to the system.	1. Use OOP. 2. Use of Glass-Box extensible mechanism. 3. Standard design architecture and standardize coding practices for any addition.
		3. have a high degree of re-use to increase the productivity of the coders, decrease future maintenance cost (see Req.10) and use crowdsourcing to fix bugs. 3.1. Shall have a high degree of Re-use in the analysis types i.e., for structural analysis all modules to re-use methodology as much as possible 3.2. Shall have a high degree of Re-use in the code.	1. OOP. 2. Modularized. 3. Creation of dedicated interfaces to 3rd party programs such as CAD, FEA, CFD.

Table 4-1 Revised Objectives and Requirements Matrix for 2nd Generation of the Software Artifact (continued)

	OBJECTIVE	**USER REQUIREMENTS** **The system shall:**	**DESIGN / TECHNOLOGICAL CHOICES**
		4. all tools developed within the system shall be easy to integrate to allow extensibility (see req. 2).	1. OOP. 2. Dynamic integration.
		5. shall allow for the injection of judgement and creativity of the engineers as PMDO is not meant to be a push button solution but rather one that helps engineers as 'what-if' question holistically and quickly.	1. All processes to be centered on the user experience. 2. Allow for the constant assessment of applicable Engineering Best Practices and Standard Work.
		6. be easy to maintain.	1. OOP for data structure. 2. Creation of interfaces for 3rd part programs. 3. Standardization of coding practise. 4. High degree of re-use.

Table 4-1 Revised Objectives and Requirements Matrix for 2nd Generation of the Software Artifact (continued)

Gen.	OBJECTIVE	USER REQUIREMENTS The system shall:	DESIGN / TECHNOLOGICAL CHOICES
		7. facilitate geometric representation of components 7.1. shall model effectively (allow for the design of over 90% of current rotors). 7.2. shall have highly efficient and simplified CAD parameterization to reduce execution time (See Req. 1) of CAD updates and increase maintainability (See Req. 6).	1. Only features that had an effect on the ultimate architecture of the turbine were considered.
		8. increase the level of definition of the turbine rotor exiting pre-detailed design which would lead to reduction in future work in detailed design (and potential expensive rework).	1. Higher fidelity tools that previously used. 2. Perform analysis at the pre-detailed stage that would have been left to detailed design.
		9. be back-wards compatible.	1. OOP with control of loading process.
		10. be optimization capable to allow for the PMDO.	1. Use the model-view-controller methodology so that the model could be separated and compiled to be run with no user input.

4.1.1 Execution Progress

When the work of creating the artifact started, the way people would work was very clear to the team developing the tool; since different parts of the rotor geometry (airfoil, platform etc.) were dependent on each other for positioning, top to bottom, the tool was to be used top to bottom (i.e., designing the airfoil and making your way towards the shaft, platform to disc). The tool was also to be used at the highest level of fidelity possibly as it would allow the greatest benefit to be realised. It had not occurred to the author that a user would have any reason not to use the highest level of fidelity.

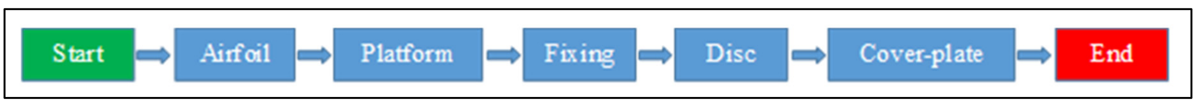


Figure 4-1 Initial Design process

It became quite obvious to the users that the tool did not offer them the same capacity they had before (i.e., the tool could not be used in part such as to design just the disc and not having to do all the other component above). With the tool built for a single highway as shown in Figure 4-1, a new way was needed that allowed the organization context to shape the artifact. There needed to be multiple on and off ramps to enter and exit the highway and so the team modified the artifact so that users could be able to use parts of it or all depending on the need.

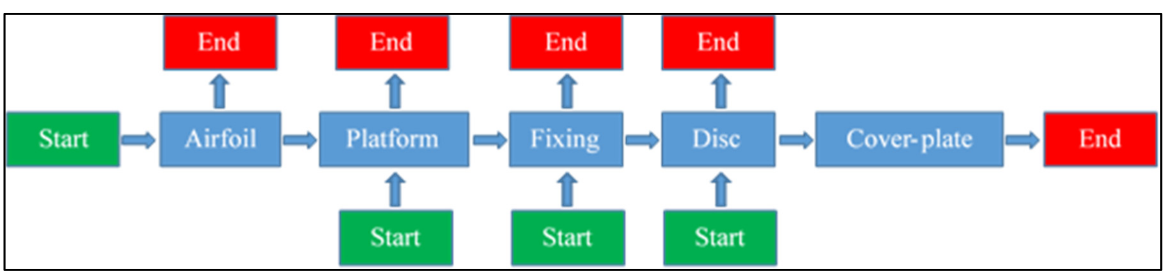


Figure 4-2 Modified Design Process

4.1.2 Engine Station and Cycle Data

Another great example where reciprocal shaping occurred is in the case of mission data. Before the launch of this project, employees at P&WC represented the engine performance data (typical missions, flight envelopes, etc.) how they saw fit; typically customizing the station numbering and performance parameter names (temperature, mass flows, pressures) how they thought would be best suited for the specific engine application. This led to a very efficient naming for each engine that perfectly suited the needs of that engine. This also meant that, from engine to engine, there was no explicit commonality in parameter naming or station naming. As part of this project, for an integrated and automated system to be able to analyse component lives (where the mission the engine endured was a critical input) a common way of extracting performance data was needed. Although there was hesitation among those who had grown comfortable with the established method(s) that was in use for years – or decades for some – many began to understand why the potential benefits necessitated this change.

The first solution presented to the engineers was to mandate all engine programs (legacy and new) to follow the SAE standard (SAE Aerospace Standard ARP AS755F) because it would facilitate the automatic extraction of the performance data. This is shown in Figure 4-3 for turbo-prop and turbo-shaft engines and in Figure 4-4 for turbo-fan engines. The standard keeps certain key stations constant (inlet to the engine {0}, first compressor inlet {20}, inlet to HPC {25}, exit of HPC {30}, exit of combustion chamber {40}, inlet of LPT {46}, exit of turbine {50}) independent of the application. As part of this solution, the key thermodynamic parameters that describe an engine cycle (temperature {T}, pressure {P}, mass flow {W}) were given standard names. This allowed for the automatic extraction of key parameters; one need only know the pertinent station and parameter to extract the data no matter the application.

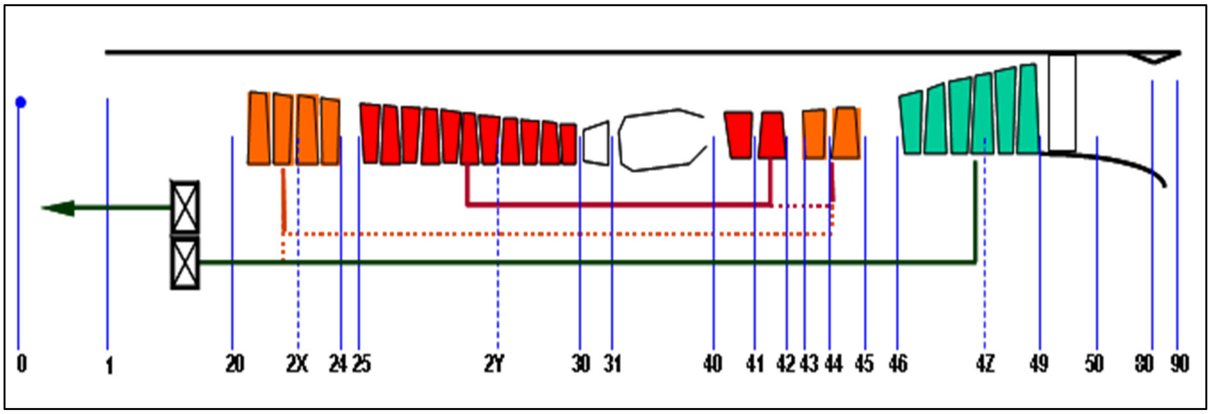


Figure 4-3 Turbo-shaft/ prop stations

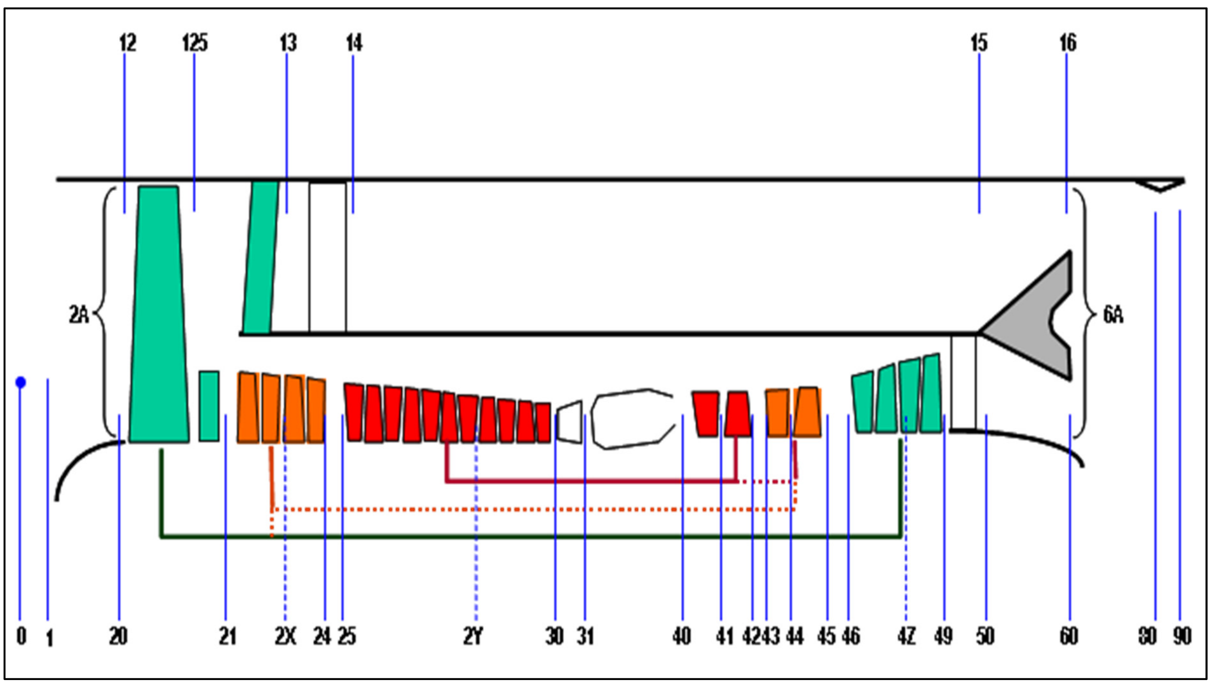


Figure 4-4 Turbo-fan Stations

This solution, although ideal in a vacuum of academic research, did not really work in the industrial environment in which it was to be implement. Too many codes and processes that had been used for sometimes decades were too ingrained and changing them to a more generic parameter and station naming would have been a gargantuan task that was not in the scope of this project. This further illustrated the pitfalls of a collaborative project between industry and academia, a solution that works in isolation does not work when considering the organization requirements and limitations.

The second solution was to add a layer to interpret the legacy nomenclature while at the same time mandating future program to establish and use a standard common naming convention. This solution gave birth to the EDS module shown in Figure 0-10. Earlier iterations of EDS were created through the work of multiple student (Maria Garcia Carrasco and Antoine Grenier) before being finalized by the author.

EDS was integral to achieving an integrated and automated system that is able to analyse component lives as it aggregated the mission the engine endured and provided a common way of extracting performance data when needed by the iRSO module (module responsible for the calculation of stress and lives). Key to the effectiveness of EDS was a lexicon. The lexicon was a dictionary that was based on what the stations were, e.g., combustor exit, and not an acronym used to describe it. The iRSO module would interrogate the EDS lexicon for the information needed (i.e. High Pressure Turbine Blade inlet Temperature) and the lexicon would be responsible of converting this request to the engine specific nomenclature used (i.e., T41) thus allowing for the automatic extraction of data and maintaining the flexibility and knowledge ingrained in the engine specific acronyms used in the performance modelling software.

Work was done by P&WC to develop and implement this naming standard that is now used. The second solution worked because it allowed the requirements of the project to be shaped by the reality of other process not exactly in the scope of the project while at the same time instituting the commonality needed by the artifact. P&WC benefited from this solution because it introduced consistency through their various engine lines, decreasing potential misunderstanding and costly rework.

4.2 Gen.1 Solutions

As any ADR process, the road to the final artifact is littered with rework. As stated in paragraph 1.2, the first stage of ADR necessitates the securing of long-term funding and commitment from the industry, and that was the main objective at the beginning. In the competitive business climate, every project needs to justify its existence year upon year. Thus, the first iteration of what was to be iRSO focused on showing not only the potential of the artifact but tangible gains in the first few years. The decision at that time was made to develop and put into production a disc only module as quickly as possible.

4.2.1 Disc only module

The disc only module achieved the goals it was set out to achieve, put something in production as quick as possible that provided utility to the P&WC as quickly as possible to justify the financial expenditure. The tool was a definite upgrade from what was there before and it can be seen from the work of Lagloire, Ouellet et.al (Lagloire, et al., 2013) and Ouellet (Ouellet, et al., 2016) as it provided great results. The tool linked together CAD, FEA and GUI through a CAE that could manage the communication. Users were able to get a higher fidelity tool that could give them automated FEA analysis in a matter of minutes instead of hours or days. Design charts were also generated from this tool which hadn't been done before. This is all to say that at that time the tool was a game changer.

Unfortunately, the tool was not extensible though. The coding practices worked for that module and that alone and could not be used for anything else and therefore could not serve as a basis for iRSO. The module was not modular, did not have a single interface to the real world and did not use a structured programming such as OOP that could be inherited by future modules. This was discovered when users started to ask for more features and more components to be added to the module but the cost of adding a module did not reduce with each module to be added but was just like starting anew. Furthermore, the future maintenance cost would have been prohibitive as each tool was like a completely new software and knowledge of one did not necessary help one in maintain the other.

4.2.2 Lessons Learned

A few lessons were learnt from this experience:

- Think about a problem as a larger set of problem – always solve one problem thinking of how the solution would be useful to solve similar or bigger problems.
- Create a data structure that allows for the incorporation future states – created the structure to include things that you know will come later and make it flexible enough to be able to incorporate things you had not thought of
- Use the common expression "code like the personal who is going to maintain it is a homicidal psychopath who has your home address" – in other words, be kind to those who will come after you. Make the code as easy to maintain as possible.

These lessons learned are very similar to the principals attributed by ADR and inspired future BIE cycles. iRSO is a better tool because of the adherence to these lessons and the principles of ADR

CHAPTER 5

FINAL ARTIFACT

The artifact created deals with all levels of the engine BOM (see Figure 0-9), from the product level and assembly level through the creation of the data structure (see chapter 5.1) through to the sub-assembly level and component level by the creation of the iRSO module. The interactions of these different modules and tools are captured by Figure 0-10, reproduced in Figure 5-1.

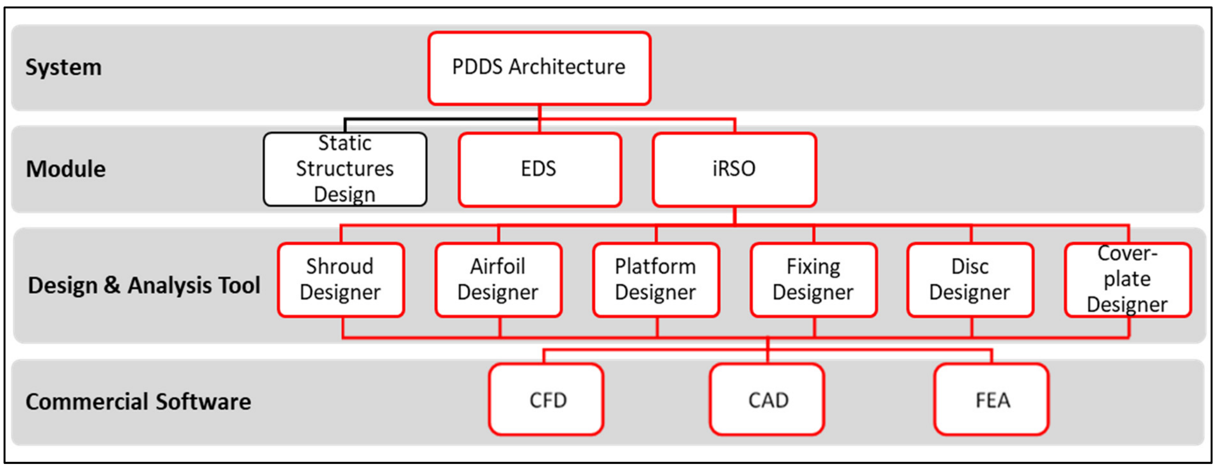


Figure 5-1 Hierarchy of Software Artifact nomenclature

The PDDS architecture at the system level underpins the entire artifact and allows for the extensibility to new modules being created for different assemblies of the engine. The iRSO module serves as the base of all the tools that are created for the design and analysis of every component of the rotor which are all aided by commercial software for specific discipline tasks (i.e., CFD for aerodynamics).

The design of the components of the rotor (airfoil, platform, fixing, disc and cover-plate) are assessed through a multi-fidelity analysis (or a series of multi-fidelity analyses) and the design altered accordingly. iRSO can be used manually where a user is able to interact with the program through various graphical user interfaces (GUIs) or in an automated fashion. It can be used to analyse a previous design or create a new one. iRSO can be used as an integrated single platform rotor designer or it can be used to design only one specific component. To best assess the artifact that was built, one must view it through the prism of the requirements it had to fulfill, these are shown in totality in Table 5-1 for both the 1st and 2nd generation of the artifact.

In this chapter, all the sub-chapters will have a reference to the requirement(s) it aims to fulfill. These requirements will be presented at the beginning of the sub-chapter in the title.

Table 5-1 User Requirements for both Gen. 1 and Gen. 2 of the Software Artifact

Gen.	OBJECTIVE	USER REQUIREMENTS The system shall:	DESIGN / TECHNOLOGICAL CHOICES
2	1. Continuously provide value to P&WC. 2. Reduce the design time of turbines by a factor of 10. 3. Reduce tools specific training needed to design a component. 4. Facilitate engineers to control the design process and to inject judgement and creativity.	1. reduce the time taken to create and analyze rotors during the design (both pre-detailed and detailed design).	1. Seamless Access to varying level of fidelity analyses.
		2. be scalable to allow an increase in the number of users as the system and tools mature.	
		3. allow inexperienced software designers and programmers to develop good software as the research team will have varying degree of experience with coding (some with none) but quality needs to be maintained.	1. Standardization of coding practices.
		4. be rapidly deployed to ensure customer satisfaction by early and continuously delivering valuable software.	1. Familiar software and analytical methodology shall be used. 2. Use aspects of Agile programming. 2.1. Close, daily cooperation between software developers and client.

Table 5-1 User Requirements for both Gen. 1 and Gen. 2 of the Software Artifact (continued)

Gen.	**OBJECTIVE**	**USER REQUIREMENTS** **The system shall:**	**DESIGN / TECHNOLOGICAL CHOICES**
	5. Have flexibility in modeling and analysis that allows the reproduction of rotor designs and creation of future designs. 6. Create a system that is scalable for future expansion. 7. Lower the risk of the designs leaving concept design phase and pre-detailed design phase.		2.2. Co-location to ensure frequent face-to-face conversations. 2.3. Simplicity in the code. 3. Use aspects of 'pure' waterfall model of distinct phase of coding and testing. 4. Tools developed independently; integration shall occur in future versions.
		5. be easy to use 5.1. shall require minimal tool specific training before one can effectively use the tool.	1. Shall have a single graphical interface that will display all inputs, CAD representation and analytical outputs independent of the tools used to generate the data.

Table 5-1 User Requirements for both Gen. 1 and Gen. 2 of the Software Artifact (continued)

Gen.	**OBJECTIVE**	**USER REQUIREMENTS** **The system shall:**	**DESIGN/ TECHNOLOGICAL CHOICES**
		6. be extensible 6.1. to allow growth of data structure so that future expansion of the data structure to include other assemblies of the engine (compressor, combustor etc.) that would incorporate other disciplines. 6.2. to allow the addition of future functionality to the system.	1. Use OOP. 2. Use of Glass-Box extensible mechanism. 3. Standard design architecture and standardize coding practices for any addition.
		7. have a high degree of Re-use to increase the productivity of the coders, decrease future maintenance cost (see Req.10) and use crowdsourcing to fix bugs. 7.1. Shall have a high degree of Re-use in the analysis types i.e., for structural analysis all modules to re-use methodology as much as possible. 7.2. Shall have a high degree of Re-use in the code.	1. OOP. 2. Modularized. 3. Creation of dedicated interfaces to 3rd party programs such as CAD, FEA, CFD.
		8. all tools developed within the system shall be easy to integrate to allow extensibility (see req. 6).	1. OOP. 2. Dynamic integration.

Table 5-1 User Requirements for both Gen. 1 and Gen. 2 of the Software Artifact (continued)

Gen.	**OBJECTIVE**	**USER REQUIREMENTS** **The system shall:**	**DESIGN/ TECHNOLOGICAL CHOICES**
		9. allow for the injection of judgement and creativity of the engineers as PMDO is not meant to be a push button solution but rather one that helps engineers as 'what-if' question holistically and quickly.	1. All processes to be centered on the user experience 2. Allow for the constant assessment of applicable Engineering Best Practices and Standard Work
		10. be easy to maintain.	1. OOP for data structure. 2. Creation of interfaces for 3rd part programs. 3. Standardization of coding practise. 4. High degree of re-use.

Table 5-1 User Requirements for both Gen. 1 and Gen. 2 of the Software Artifact (continued)

Gen.	OBJECTIVE	USER REQUIREMENTS The system shall:	DESIGN/ TECHNOLOGICAL CHOICES
		11. facilitate geometric representation of components 11.1. shall model effectively (allow for the design of over 90% of current rotors) 11.2. shall have highly efficient and simplified CAD parameterization to reduce execution time (See Req. 1) of CAD updates and increase maintainability (See Req. 10)	1. Only features that had an effect on the ultimate architecture of the turbine were considered
		12. increase the level of definition of the turbine rotor exiting pre-detailed design which would lead to reduction in future work in detailed design (and potential expensive rework)	1. Higher fidelity tools that previously used 2. Perform analysis at the pre-detailed stage that would have been left to detailed design
		13. be back-wards compatible	1. OOP with control of loading process
		14. be optimization capable to allow for the PMDO	1. Use the model-view-controller methodology so that the model could be separated and compiled to be run with no user input

5.1 Coding Practices (req.2-4, req.6-9, req.13, req.14)

With such large IT/S (information technology / information system) research teams, it is extremely important to establish and adhere to strict coding methodologies and practices. After some initial false starts with defining an architecture that could be used through the whole artifact, the team settle on the notion of using Object Oriented Programming (OOP). OOP, especially the way it was implemented in PDDS, offers many advantages over alternatives:

- It allows for management of compatibility – i.e., all files saved on old version of the tool would be compatible with relatively little work.
- The class framework gave rigidity to code – a very important aspect when research team members had varying degree of programing skills.
- The concept of inheritance allowed for the creation of super classes – minimizing rework for each new team member.
- It allows for a single interface to exist between the system and commercial software
- It allows for the centralization of data.

The first steps in establishing the structure were to create core classes that the whole team could use and build from. These core classes were a component class that everything could inherit from, a CAD interface that was used to communicate with the CAD software (catiaBasedGeometry) and an FEA interface (ANSYSInterface) to communicate with the FEA program. The component class implemented these two interfaces and contained all the parameters through the container class (parameterSet). A unified modeling language (UML) of these core classes is shown in Figure 5-2. It worth noting again that due to potential of this research to give Pratt and Whitney a large industrial advantage, some details are withheld due to its propriety nature.

In all the UMLs a few items are key:

- G indicates that it inherits from a pass by reference abstract class – class that keeps only one copy of the object irrespective of how many times it is reassigned in order to avoid unnecessary copies
- V indicates that it inherits from a value class – a class that creates an independent copy of the object every time it is assigned.
- An arrow with an empty head indicates inheritance and the direction of the arrow indicates the super class.
- An arrow with a solid head and a diamond tail indicates aggregation where the direction of the arrow indicates the 'has a' relation – in Figure 5-2 for example, a component has a parameter set. The number at the head and tail of the arrow indicates how many of that class are contained. In the same example in Figure 5-2, the 1 and 1 indicates that a single component contains a single parameter set. In Figure 5-4 where we see 1 at the tail of the arrow leading from the turbine to the stage class and an 'n' in the head, indicating that a turbine may have 1 or multiple stages.

These core classes are used to build the complete architecture. In Figure 5-3, one can see that every component inherits from the core component class.

In Figure 5-4, once can see the sub assembly and assembly levels of the engine BOM, see Figure 0-9, coming to life. Each turbine contains 1 or multiple stages which have the rotor and stator sub-assemblies. Figure 5-5 shows how these various sub-assemblies all fit in to create the different assemblies of the engine and how all the assemblies create the final product (the engine). This architecture closely resembles the engine BOM illustrated in Figure 0-9 with a few extra layers added to contain some general knowledge shared by different members of the sub-assembly or assembly. As the figures show, the data structure is extensive in that it considered all aspects of engine design, even those that have not been fully implemented at the writing of this thesis.

The biggest benefit of the usage of OOP and inheritance as implemented in the UML diagrams is that it greatly facilitates scalability of the data structures to assemblies that will be the work of future researchers (such as compressor). Through the inheritance of the component class, any sub-assembly or component created under the compressor will have access to the basic functionality of CAD interfacing, FEA and CFD interfacing as well as a way to handle all parameters involved. This undoubtedly will make future additions to the artifact much quicker and easier to implement.

The idea of implementing OOP is one that Maxime Moret (Ph.D. student with PDDS) and the author both had at the onset of generation 2. Of the framework. The implementation was mostly done by the author with some functions written by masters and bachelors students under the author's strict supervision.

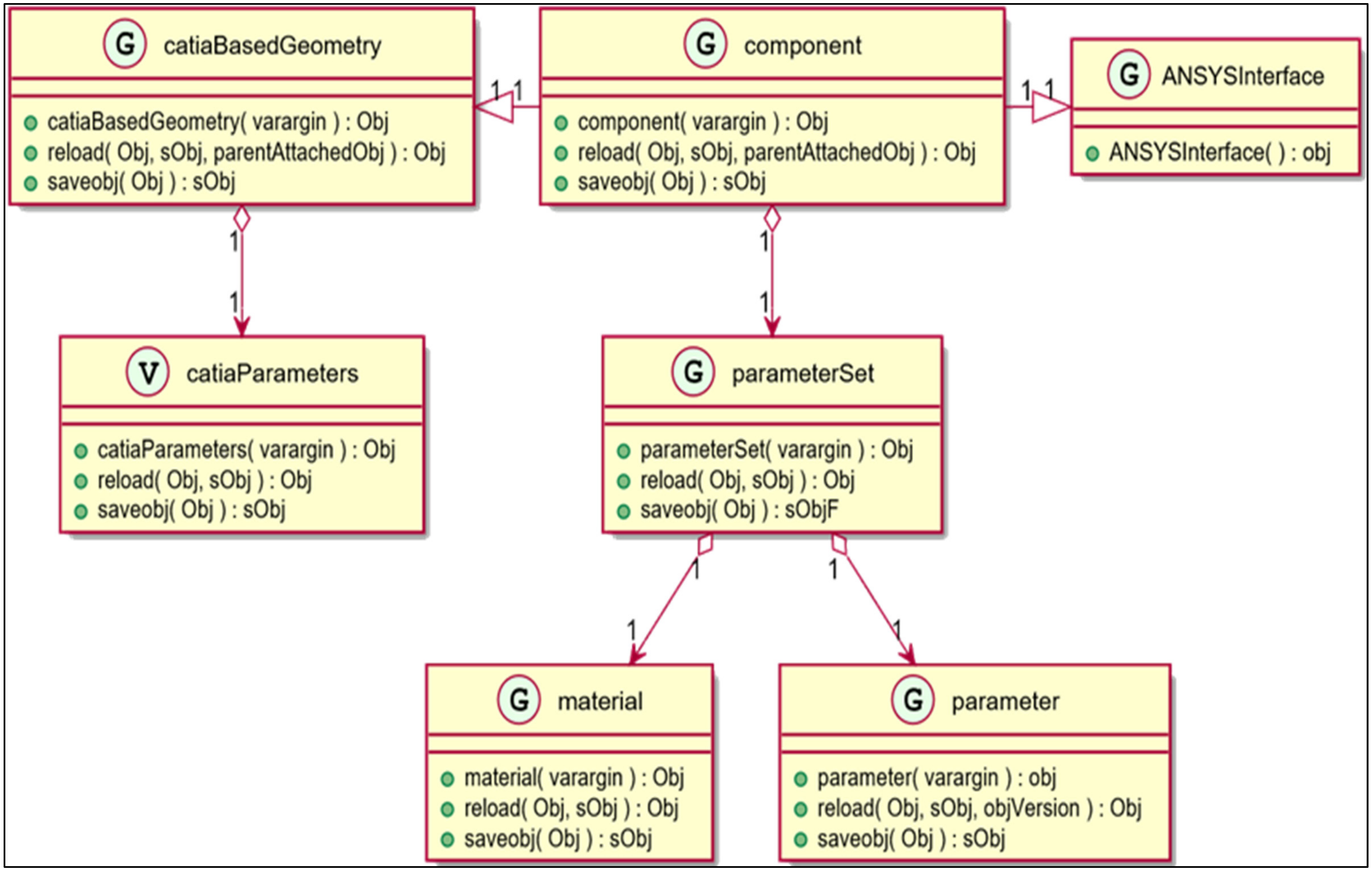


Figure 5-2 UML - Core Classes

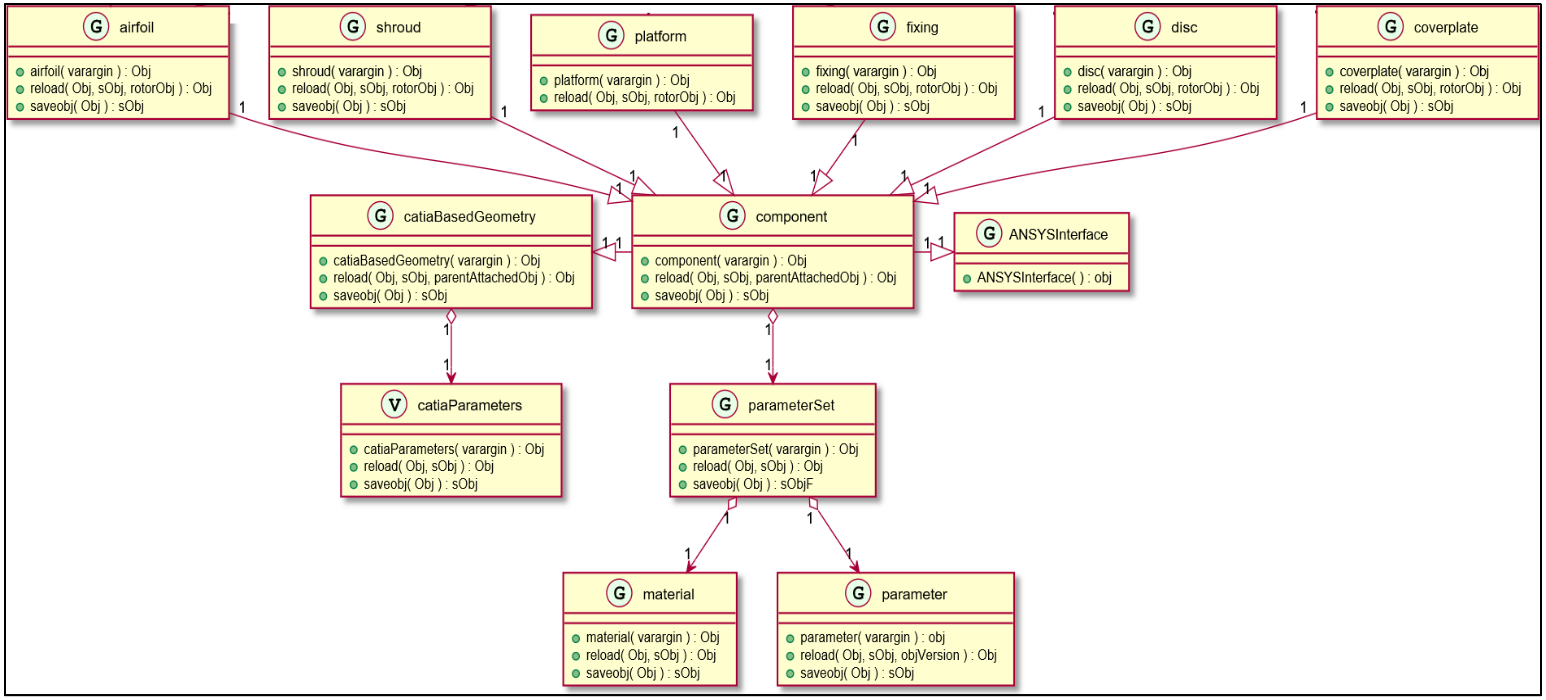


Figure 5-3 UML – Component Inheritance

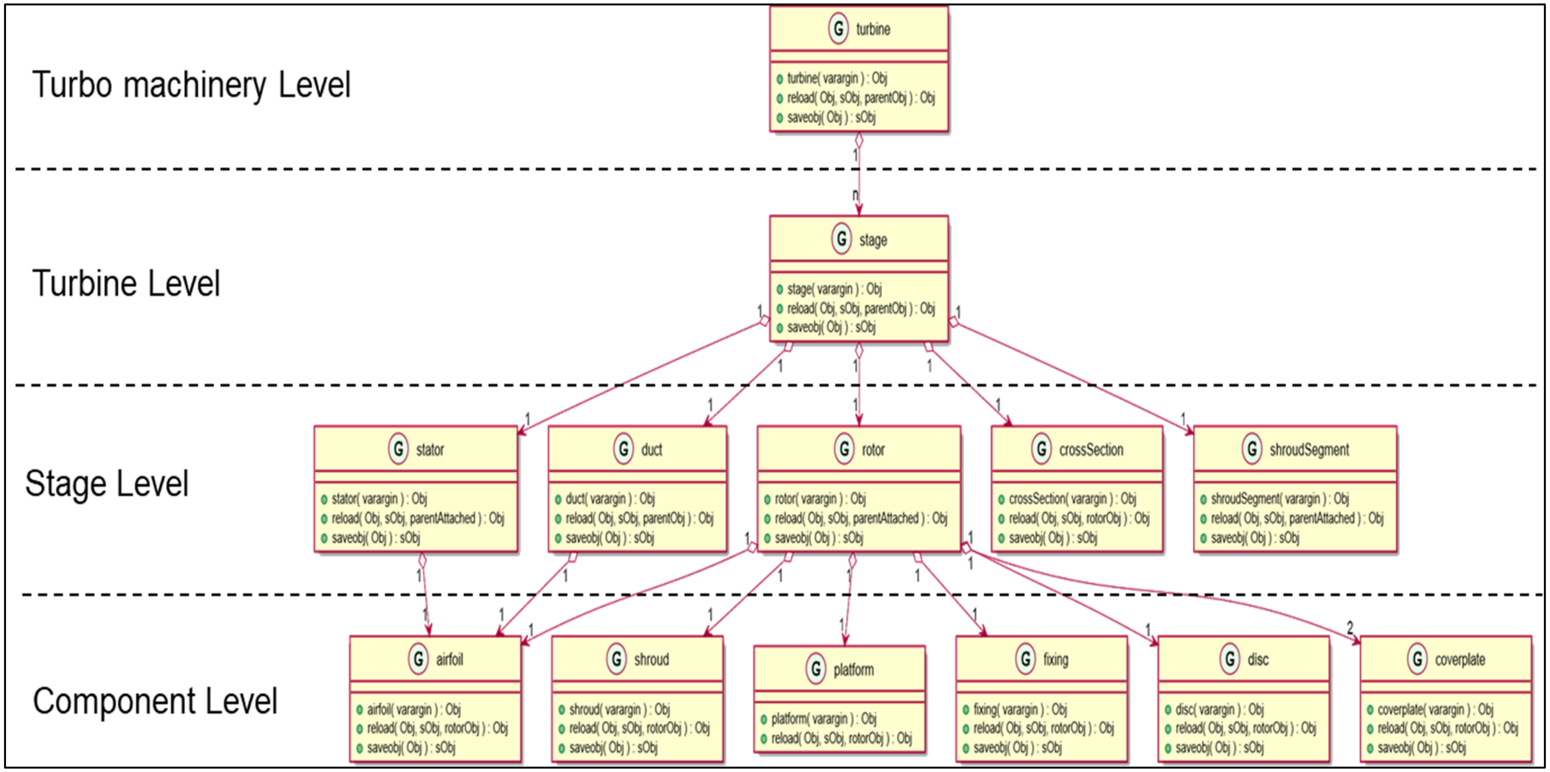


Figure 5-4 UML- Turbo machinery Level

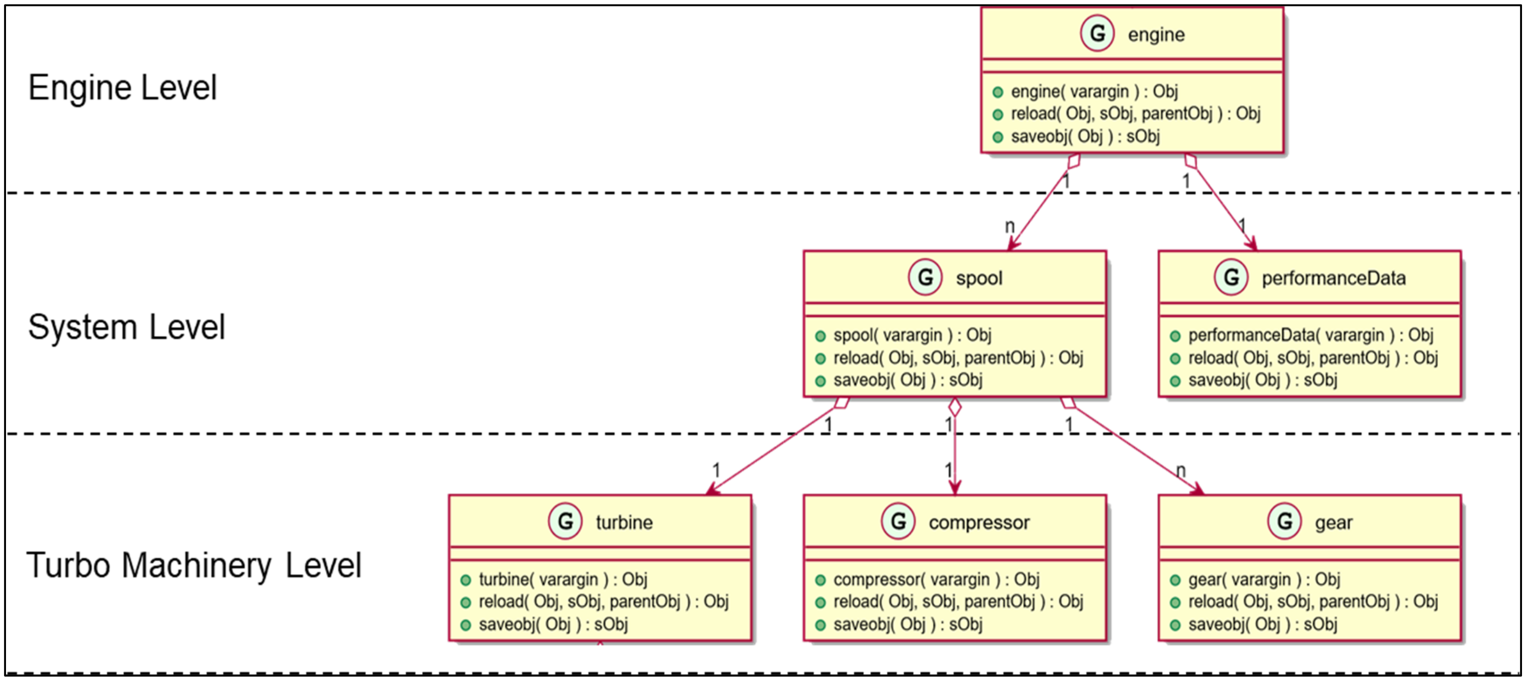


Figure 5-5 UML - Engine Level

5.2 CAD – FEA – CAE Software Relationship (req.2-3, req.6-9, req.13, req.14)

As mentioned in sub-chapter 5.1, one of the key benefits of using OOP was that it allows the unique method of handling all interaction with the 'outside world'. Previous work done, which did not implement OOP, by Lagloire et. al. (2013) in the gen.1 of the artifact, had defined software and architecture that would allow the CAE software (user interface) to communicate with CAD software and analysis software. As shown in Figure 5-6, the CAE software interacts to the CAD through a gateway program (Transfer 1) and vice-versa (Transfer 2). The interaction to FEA software is handled though a different gateway (Transfer 3-6). Each transfer illustrated in Figure 5-6 does a specific job.

- Transfer 1 is responsible for the sending inputs the CAD software and updating the parametric model with those variables facilitated by the gateway program.
- Transfer 2 is responsible of obtaining any geometrical data necessary (such as areas, lengths, inertia) from the CAD software to the CAE software facilitated by the gateway program.
- Transfer 3 is the creation of an analysis file for use in FEA analysis.
- Transfer 4 generates a geometry file in a format needed by the FEA package for the use in FEA analysis. The transfer is facilitated by the gateway program.
- Transfer 5 is the creation a command file for the FEA package that tells the FEA package what commands to call in sequence.
- Transfer 6 retrieves the FEA analysis data such as stress and temperature values at required position.

This architecture allows the use of specialised software (BCA approach) to handle the tasks it was best suited for and left execution control and user feedback to the CAE software. The work of Lagloire et. al (2013) did not however have a unique way it implemented this execution control in the CAE software across all future tools (at the time of the work of Lagloire et. al. (2013) only the gen.1 version of disc tool was completed) of the iRSO module. In the gen.2 of the iRSO module, this architecture was implemented in the catiaBasedGeometry and

ANSYSInterfaces classes as shown in Figure 5-2 thus allowing the execution control to be applied uniquely throughout the entire artifact. Due to potential of this research to give Pratt and Whitney a large industrial advantage, some details are withheld due to its propriety nature.

As mentioned before, the work by Lagloire et. al. (2013) was pivotal in defining the gateway program and therefore Transfers 1, 2 and 4 but didn't implement them in a way that was extensible. In gen. 2 of the artifact, the author implemented these transfers in the extensible interfaces: catiaBasedGeometry and ANSYSInterfaces. Transfers 3, 5 and 6 were also implemented at that time by the author.

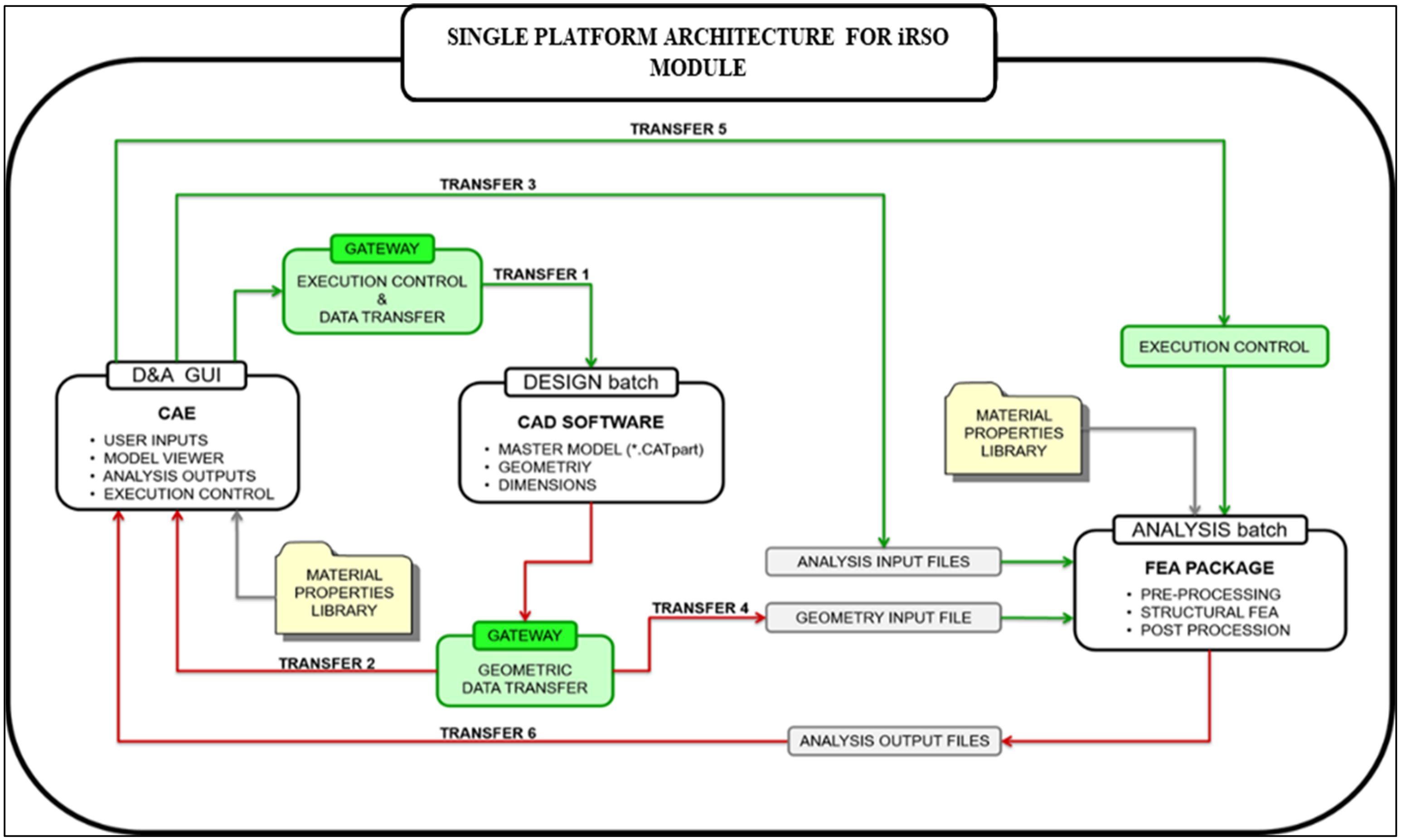


Figure 5-6 Architecture for Interaction of iRSO with the Outside World

73

5.3 CAD Parameterization (req. 5, req. 10-12, req.14)

After the architecture, the next thing of critical importance is the methodology used to represent the geometry as it allows for the:

- Artifact to be easy to use with simple and effective parameterization.
- Increase in the level of definition of the turbine rotor exiting pre-detailed design which would lead to reduction in future work in detailed design (and potential expensive rework).
- Optimization of rotors through the robust and highly flexible parameterization.

Because most engineers in the field, and more importantly most engineers at P&WC, are familiar with CAD software, in this work parametric models created in CAD software are utilized as the base of the artifact. The parametric models are a basic framework of the component to be designed that could morph into a vast number of different designs depending on the value assigned to a parameter, very much like how any rectangle could be generated by changing its length and width. The parameterized models used were reported by Lagloire, et al., (2013) and. Oulett, et al., (2014) (disc geometry), Twahir et al. (2014) and by Moradi, et al., (2015) (airfoil geometry), a summary of which are presented in this section.

The CAD parameterization was where the author relied most heavily on the experience of P&WC engineers. The author aided in the creation of the fixing parmeterized model and the airfoil model and helped to update other models (shroud and disc) under supervision from P&WC engineers such as Dan Lecuyer. The work will be presented component by component to create the rotor as illustrated in Figure 5-7.

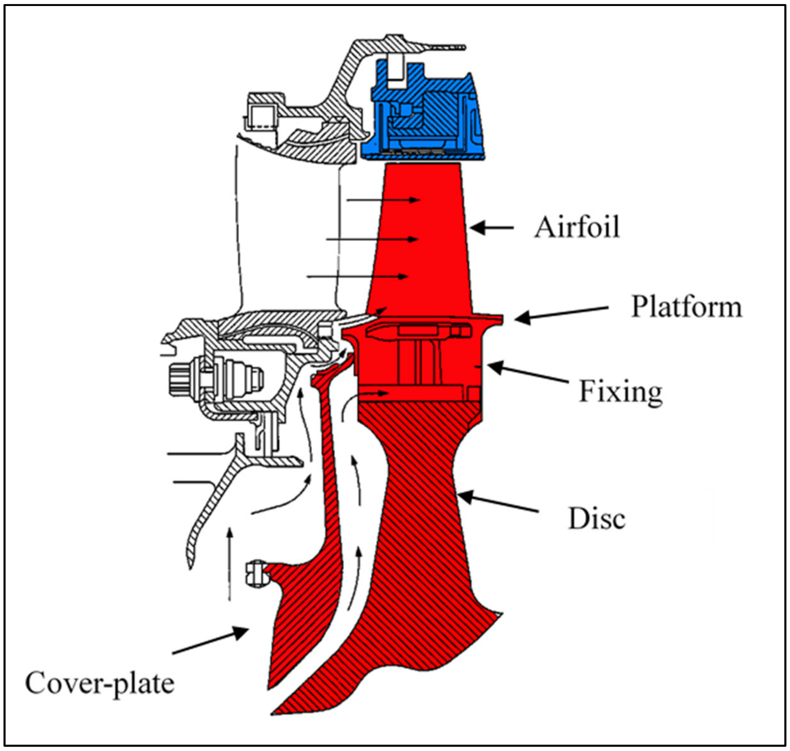


Figure 5-7 Representation of a Rotor

As mentioned in chapter 3.1, numerous minor BEI cycles were devoted to defining the best CAD parameterization of each component of the rotor (refer to Figure 0-9 for what is defined as a component in this work). Some information is given on some of these cycles in CHAPTER 4 but due to potential of this research to give P&WC a large industrial advantage, some details are withheld due to its propriety nature. Some details of the parameterization, such as the total number of parameters needed and the definition of all the parameters is also withheld for the same reasons.

5.3.1 Airfoil

The airfoil parametric model is a multi-section exterior surface where each section defines a desired radius chosen to be a design section. At the design sections, the key leading and trailing edge parameters such as diameter, metal angle and wedge angle are defined, as well as the axial chord and stagger angle. The throat opening defines the length of the throat and the uncovered turning (UT) defining how much turning occurs after the covered section of the passage (i.e., portion where the flow is not constricted by walls on either side of it).

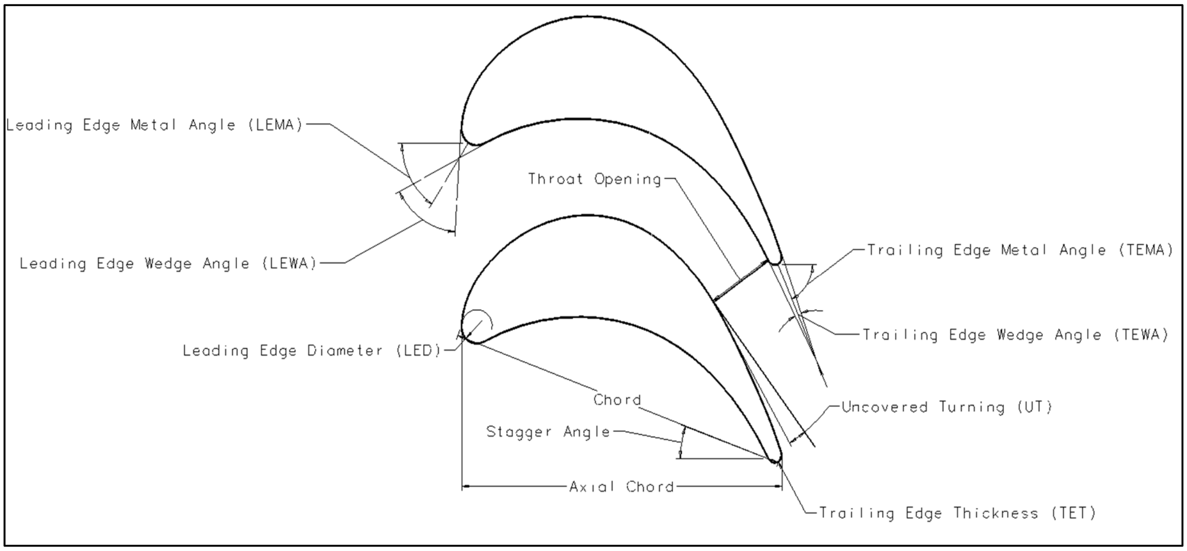


Figure 5-8 Airfoil Parametric Model - Section Parameters

In the case where an airfoil requires a pocket for either structural reasons or dynamic or even to model space for cooling flow, an extra section is defined in the interior of each of the design section. This pocket, much like the exterior surface, is defined by a set of parameters such as thickness to the pressure and suction side and the depth of the entire pocket.

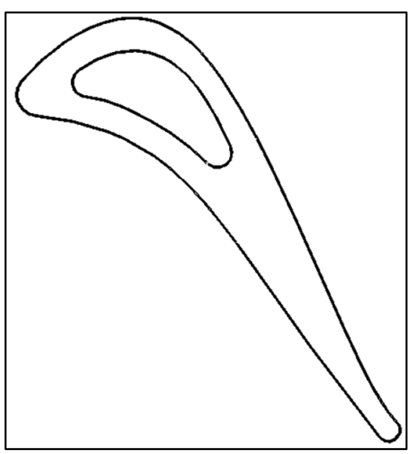


Figure 5-9 Airfoil Parametric Model - Pocket definition

The area between individual sections is defined by interpolating along the radial direction between the sections through a fourth-degree spline passing through key parts of each design section. Each design section can further be translated axially and tangentially to affect the final 3D airfoil. A fully defined airfoil may then look similar to Figure 5-10:

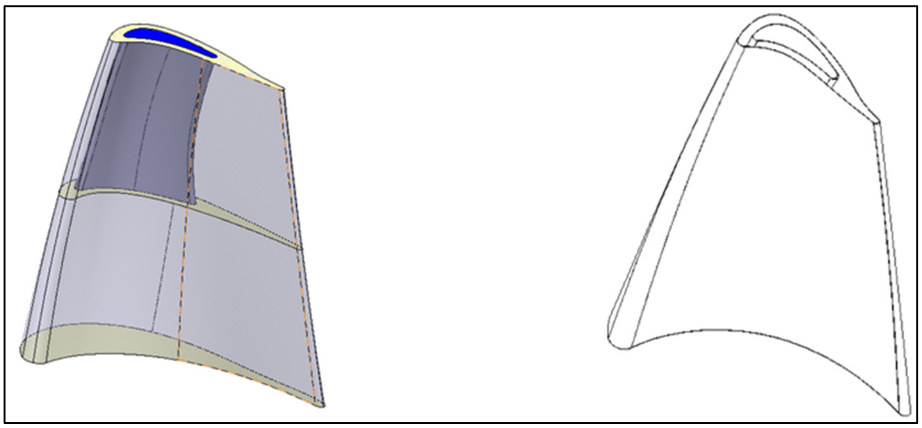


Figure 5-10 Airfoil Parametric Model - 3D Airfoil

5.3.2 Platform

The platform, the portion of the rotor that defines the hub gas path, model starts with the modelling of the cross-sectional view of it. The airfoil hub dictates the gas path portion of the platform with the hub radius, flare angle and axial chord being the parameters that create the initial schematic. Varying thickness of walls at the leading edge, trailing edge and at the gas path together with the location of the overhangs define the rest. This is shown in Figure 5-11

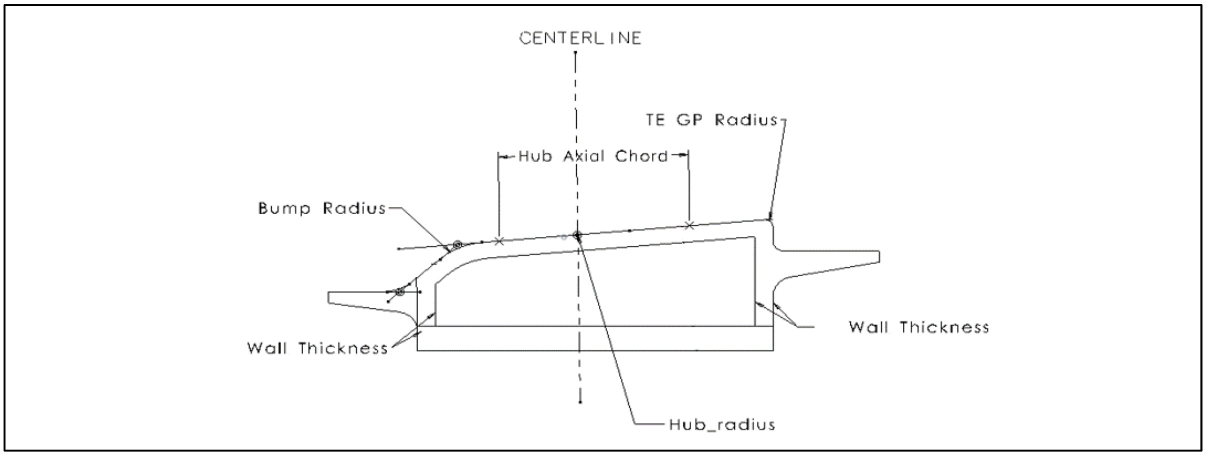


Figure 5-11 Platform Parametric Model

Tangentially, the platform lies on a cylinder defined by the hub radius and its tangential width limited by the number of blades. Its width is the circumference of a sector of a circle defined by how number of blades (NOB) are on the disc. This is shown below by the multiplication of the circumference of a circle ($2\pi r$) multiplied by the ratio between the angle of the sector ($\frac{2\pi}{NOB}$) and the angle of a full circle (2π).

$$platform\ width = 2\pi r * \frac{\frac{2\pi}{NOB}}{2\pi} = \frac{2\pi r}{NOB} \quad (1)$$

5.3.3 Fixing

For this work, only 1 to 4 lobe fixing were parameterized as shown in Figure 5-12, each having its own seed model.

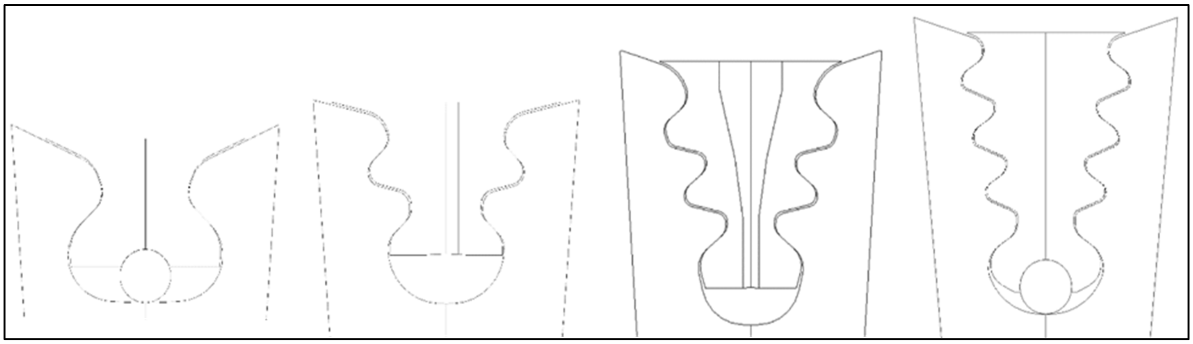


Figure 5-12 Fixing Parametric Model - 1 to 4 Lobes

A lobe is defined by a neck and lobe pairs and is controlled by a high arc and low arc connected by a line as shown Figure 5-13

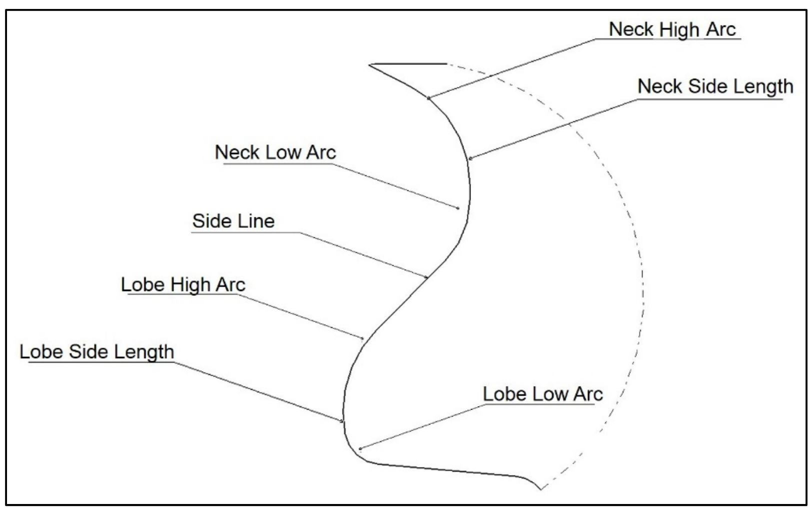


Figure 5-13 Fixing Parametric Model - Neck and Lobe Pair

These lobe-neck pairs are joined together by an unload line segment, as shown in Figure 5-14

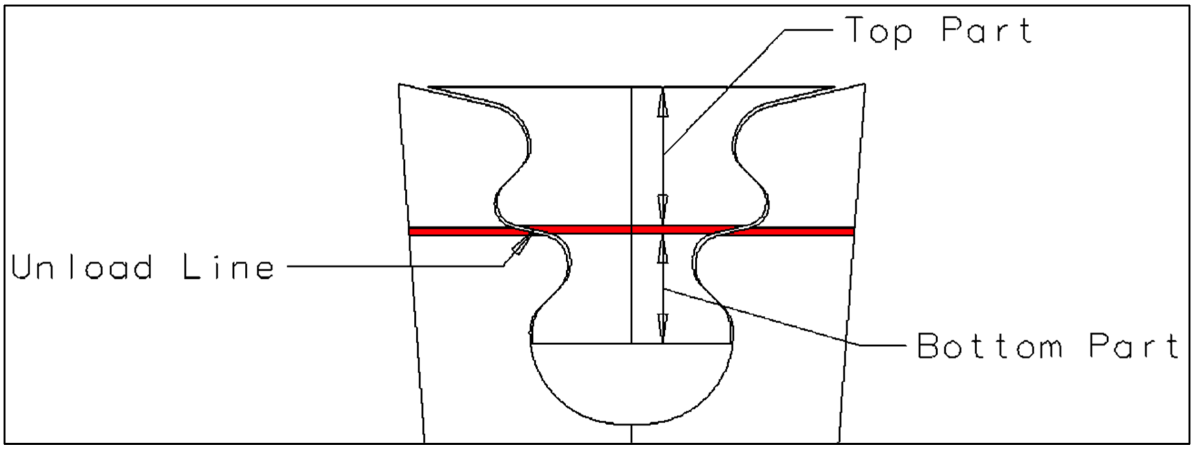


Figure 5-14 Fixing Parametric Model – Unload line

With each lobe-neck pair having identical parameterization, the increased number of pairs leads to more parameters; the 2 lobe having more than the one and even more for the 3 lobe and 4 lobe.

For cooled airfoils, feed passages start at the bottom of the fixing. The total cooling area of the blade fixing is calculated by an experienced engineer and is used to start the definition of the cooling passages. The wall thickness, usually a parameter that's limited by manufacturing considerations, and the total number of cooling passages required further define the bottom of the fixing. This is illustrated in Figure 5-15.

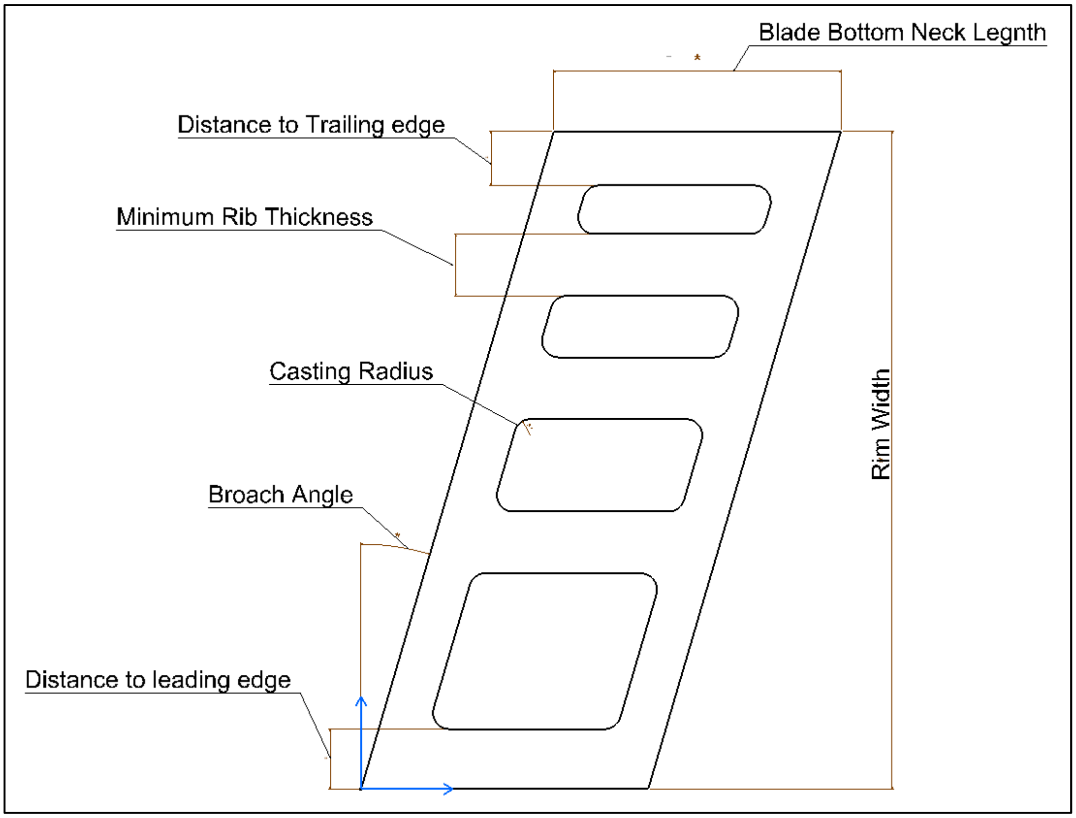


Figure 5-15 Fixing Parametric model - Cooling Passages

5.3.4 Disc

Being axisymmetric allows the disc to be wholly parameterized on the axisymmetric plane. With the disc being the lowest component, its height is restricted by the position of the bore and the ending of the fixing. The shape is dictated by three necks (high, mid and low) with the highest joining the fixing width and lowest the bore. This is shown in Figure 5-16

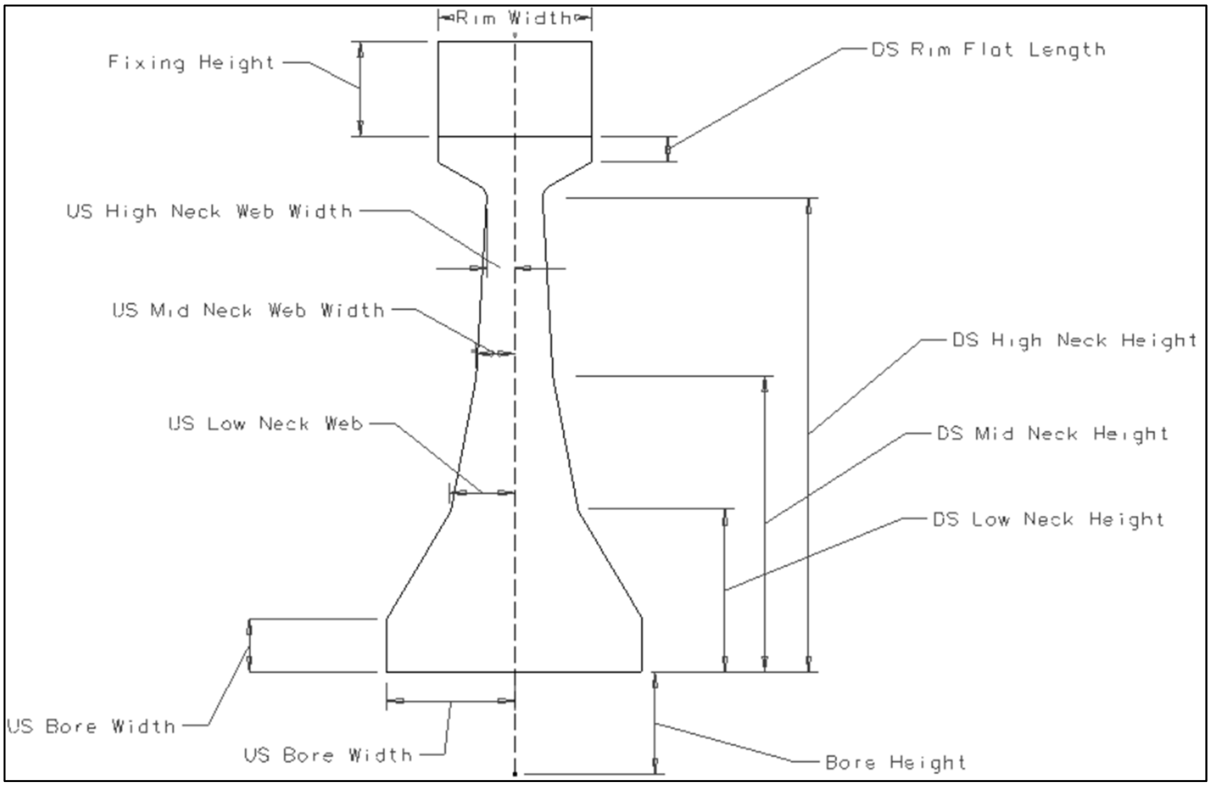


Figure 5-16 Disc Parametric Model

5.3.5 Cover-plate

The cover-plate parameterization is tied to, as one would expect, the shape of the side of the disc it`s closest to. A skeleton is created from offsets from the side of the disc and vertically from the bore radius. The outer shape is then offset from this basic skeleton to give it thickness. This is shown in Figure 5-17. This parameterization is independent of whether the cover-plate is upstream of the disc or downstream.

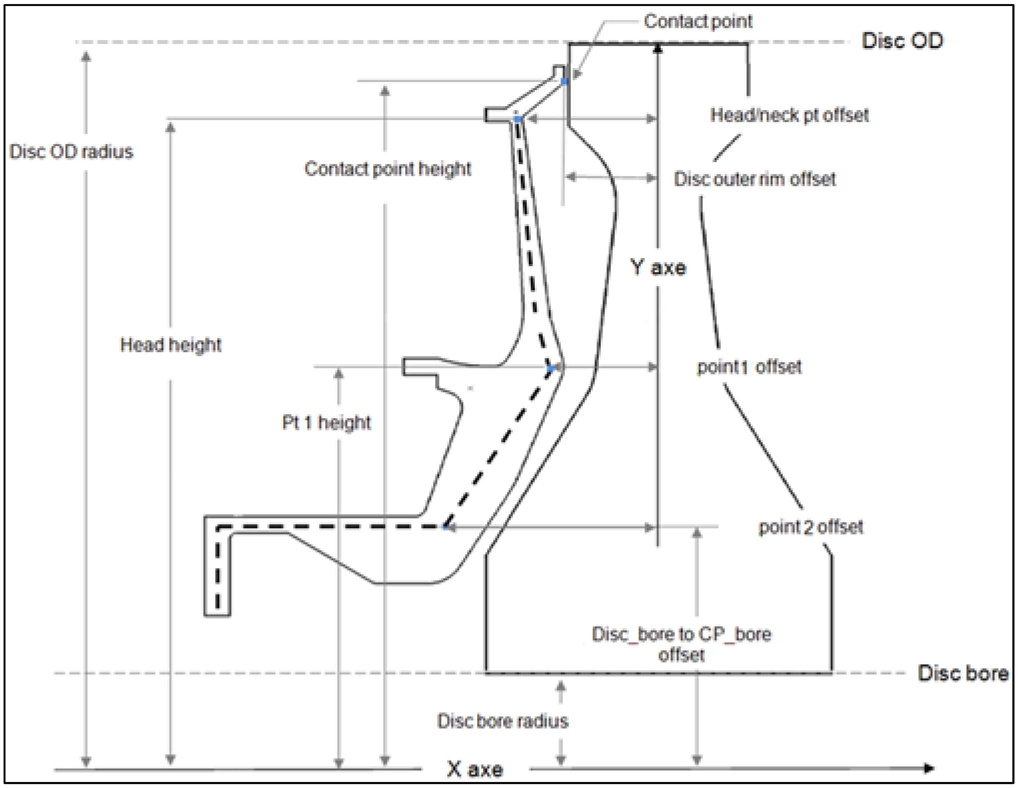


Figure 5-17 Cover-plate Parametric Model

5.4 Multi-Level Fidelity Tools (req.1)

Incorporating numerous disciplines into a single design system means that a lot of processes will need to be run. The author uses component centric feasibility sequential approach (CC-FSA) in iRSO, meaning that it understands that an airfoil sits on the platform which is on the fixing and so on to the disc, and therefore these components are geometrically linked to each other; necessitating the need for the FSA. Additionally, it is well understood that the design of rotors is multi-disciplinary in nature; excluding any one discipline or considering only one discipline at a time when designing a component results in sub-optimum design, rework and expensive delays. Figure 5-18 shows CC-FSA where at each level of component design, the multiple disciplines that support it can be engaged simultaneously to assess its characteristics. A component optimization happens by assessing the results of each discipline’s analysis and the engineer can assign weighted importance to each disciplines requirement (depending on the goal of the rotor to be designed) to come up with a component that has been designed and optimized multi-disciplinarily.

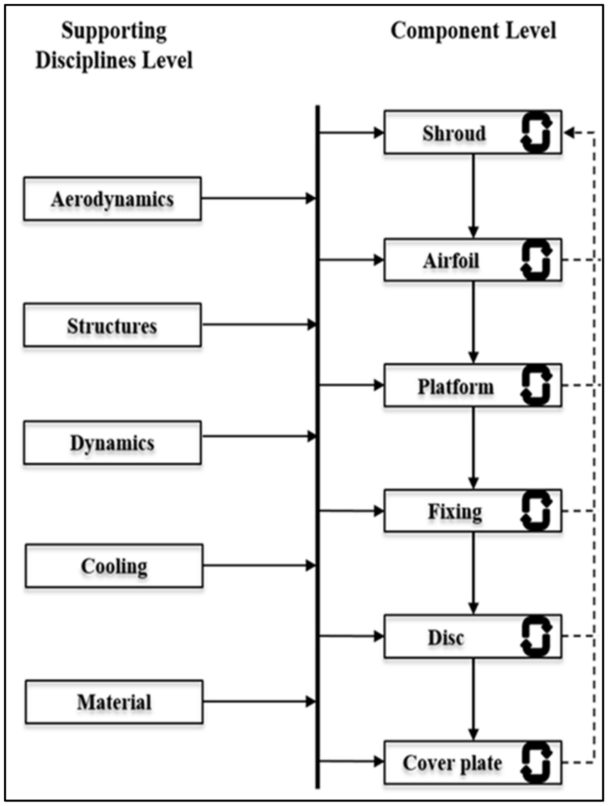


Figure 5-18 Representation of the CC-FSA

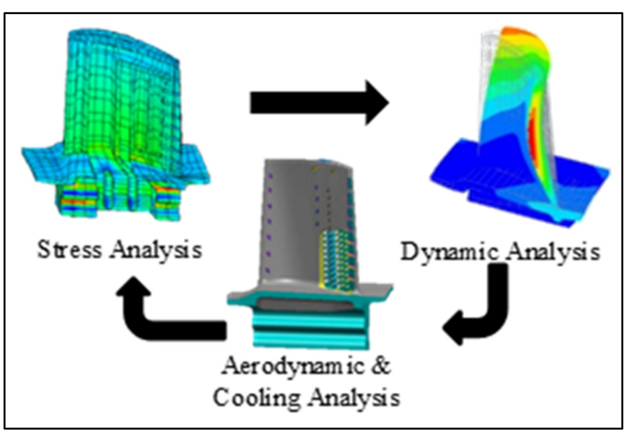


Figure 5-19 Multi-Disciplinary Optimization Loop

As can be seen in Figure 5-19 with the example of the blade, the structural, aerodynamic and dynamic characteristics of each blade are analysed concurrently to achieve a blade that is not optimized for a discipline but offer the best compromise of all the disciplines.

This CC-FSA approach would mean that many analyses need to be run at any given iteration, resulting in a potential for large run times. To mitigate this potential problem and achieve the requirement of reducing the time taken to create and analyze rotors by 10X during the design (both pre-detailed and detailed design) (req1 and req.8) a multi-level fidelity methodology is employed.

Uni-dimensional (1D) fidelity (only considering variations in only one direction), 2D (radial and axial) fidelity as well as 3D results was deployed within the tool. When a 2D results is obtained, it is then compared to the 1D results and a factor (Kt 1 →2) calculated. The same is also done when 3D results are obtained, creating a Kt 2 →3 factor. These factors are used to predict future higher fidelity runs. The 1D results shall be calibrated by Kt 1 →2 factor, allowing 1D to predict 2D results just as 2D shall be calibrated by Kt 2 →3 factor. The product of these factors shall allow 1D results which are much faster to obtain to predict eventual 3D. This is shown in Figure 5-20

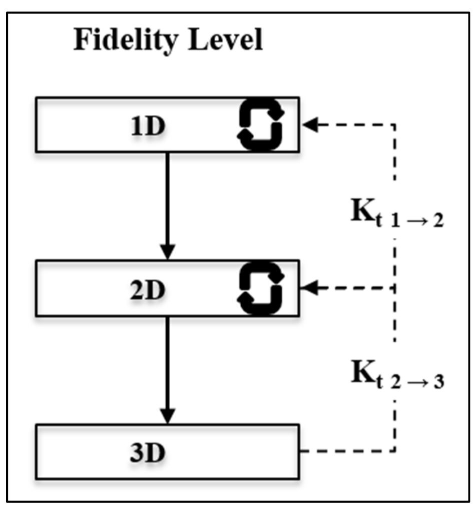


Figure 5-20 Schematic Representation of Multi-Level Fidelity

With the higher levels of fidelity only used for the calibration of the lower levels of fidelity analysis, the time taken to reach a solution could be drastically reduced. As an example, Figure 5-20 shows that if a higher-level result is used to calibrated every three runs of the lower-level fidelity analysis, many more designs could be assessed in a shorted time.

This thought process was also be used when it comes to obtaining targets. For example, a 3D stress target could be set based on testing and experience and the Kt 2 →3 factor used to get the 2D targets. Equally, the Kt 1 →2 factor could then be used to generate 1D targets.

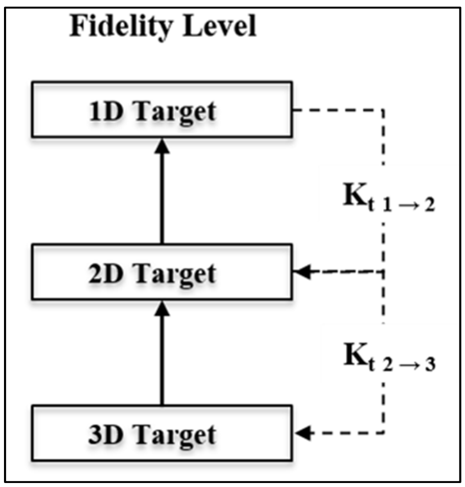


Figure 5-21 Schematic Representation of Multi-Level Targets

Another great time saver born out of the multi-fidelity approach is that the analyses done in 2D in iRSO have a high degree of correlation between the structural results obtained in iRSO and those from a 3D detailed design P&WC process therefore allowing for the $K_{t\ 2\text{->}3}$ to be stable. As shown in the work of Twahir et al (2013) and Ouellet et al (2014), the stress levels of those calculated in iRSO 2D are comparable to those found in 3-D. As can be seen in Figure 5-22, the peak stresses are both captured accurately where the maxima and minima are located in similar locations and so is the overall level of stress. Figure 5-22 shows the two stress levels where the same contour color indicates the same level of stress. In both images in Figure 5-22, one can see that the peak stresses are on the lobes of the disc fixing and the maximum stress occurring below the last lobe neck.

Due to the propriety nature of this research and its potential to give P&WC a large industrial advantage, some details have been withheld.

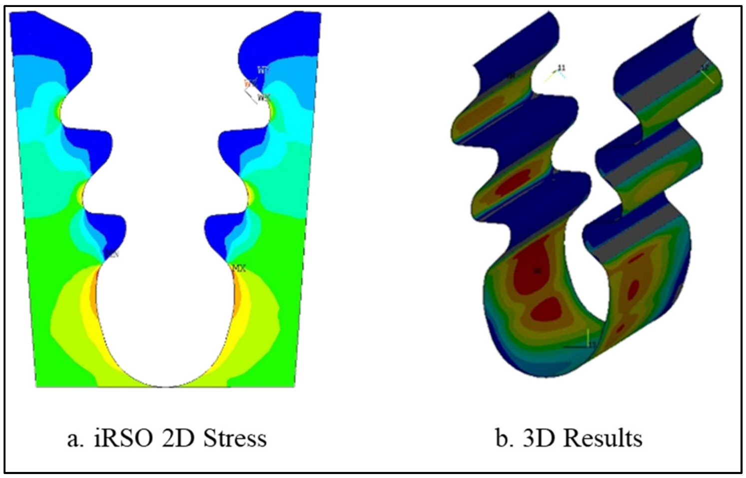


Figure 5-22 Comparison of Structural Results

The aerodynamic discipline's difference between what iRSO can offer and what detailed design tools can offer is even smaller. The same software used in detailed design for meshing, solving the CFD solution and post processing is used in iRSO. The only difference one might be able to observe is caused by the few simplifications used in iRSO to model the airfoil (iRSO has a limited number of design section and also does not offer the level of granularity when designing those design section that detailed design tools offer). As can be seen from Figure 5-23, the overall flow is very similar, with the Mach one lines appearing at the same location and the shock structure looking very similar.

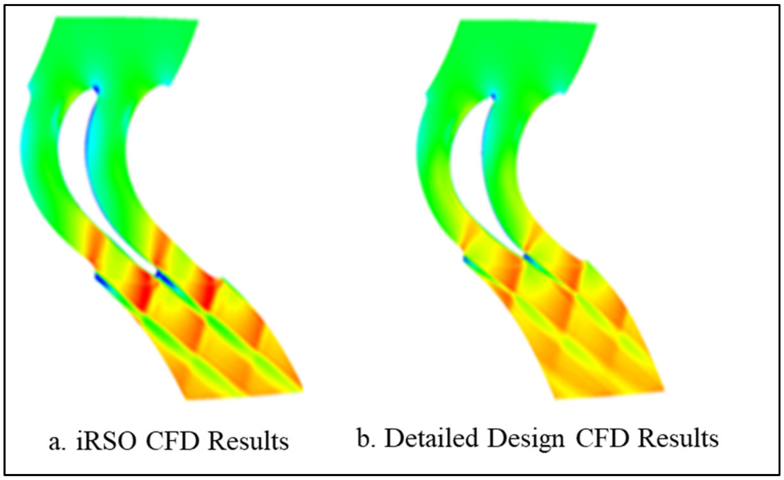


Figure 5-23 Comparison of CFD results

Ultimately iRSO can be summarized as a tool that has enough accuracy when compared to detailed design methodology and tools to allow engineers to ask 'what if' with respect to turbine rotor and receive answers that they can trust at a significantly reduced time (80% reduction in pre-detailed design time and around 30% reduction of detailed design time) than what was capable previously.

All items discussed in this chapter were implemented by the author as part of the Ph.D.

5.5 Process Representation with communication between all tools, data, and engineers (req. 5, req. 9)

As stated in literature, the most important objective of the research, "from an end-user' perspective, is the ability to interact with the process once it has been submitted" (NATO RTO Research Task Group AVT 093, 2006). As said by Sobieszczanski-Sobieski & Hafka:

> "[P]MDO , emphatically, should not be used as a push-button design procedure. The engineering design process moves forward by asking "what if" questions and using the answers to make design changes" ... "the human interface is crucially important to enable engineers to control the design process and to inject judgement and creativity" (Sobieszczanski-Sobieski & Haftka, 1997)

In the software artifact the injection of judgement and the monitoring of progress (req. 10) by the engineer shall come in the form of GUIs. The user will be able to both monitor the progress of the system as well as alter parameters to affect change in the geometry when the system is used in manual design mode. The GUI also offers the user insight into the analysis execution, providing meaningful error messages which, pinpoint the location(s) in the execution process where error(s) were encountered. This makes the artifact extremely user-friendly (req. 6).

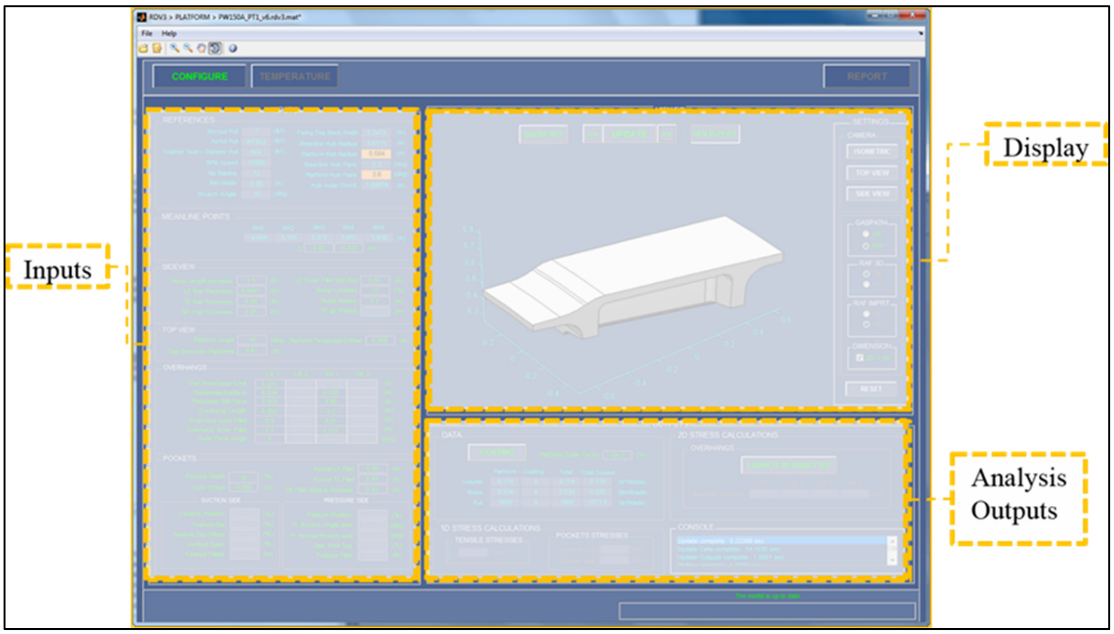


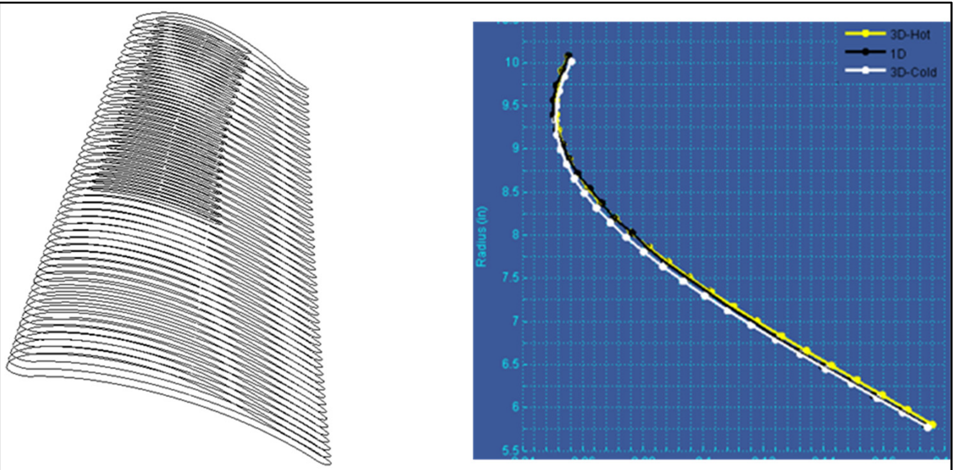
Figure 5-24 An example of GUI of a component tool in the iRSO Module

5.6 Airfoil Optimization (req.14)

5.6.1 Structural Analysis

The airfoil optimizer is an example where the engineer is able to ask the what if question in a multi-disciplinary fashion. It simultaneously allows the assessment of the airfoil from a structural and aerodynamic perspective.

Structurally, the airfoil is cut along its span, at discrete location corresponding to every 1%, 2% or 4% of span depending on the level of granularity needed. The area and perimeter of each of these sections are assessed to give a span wise distribution. This is shown in Figure 5-25.



a. iRSO Section Cuts   b. Area Distribution

Figure 5-25 Airfoil Structural Analysis – Area

With the area known it is then possible to calculate the 1D mean stress of the airfoil

$$\sigma = \frac{CF\ Load}{Area} \tag{2}$$

Where:

σ : the stress of the specific section

CF load: the centrifugal load above a specific section

Area: the net geometric area (if a cavity exists, net area is exterior surface area minus cavity area) for that section

CF load is the mass (m) multiplied by the radial distance of the center of gravity of a section (R_{CG}) multiplied by the rotational speed squared:

$$CF\ Load = m * R_{CG} * \left(\frac{RPM}{R_{CG}} * \frac{2\pi}{60s/min}\right)^2 \tag{3}$$

Mass is density (ρ) multiplied by the volume (V)

$$mass = \rho * V \tag{4}$$

Volume is the trapezoid's area

$$V = {}^{1}\!/_{2} * (A_0 + A_1) * height \tag{5}$$

With the length one (A_1) being the area above the section and length two (A_0) being the section area. The height of the trapezoid is dependent on the span and the number of cuts taken.

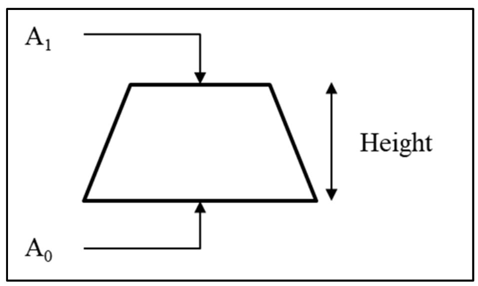


Figure 5-26 Calculation of Section Volume

With some airfoils required coating and even a thermal barrier coating (TBC) on top of that (shown in Figure 5-27), the difference in their densities is also taken into account. The perimeter of each section is multiplied by the thickness to get the volume which is then multiplied by the density to give mass.

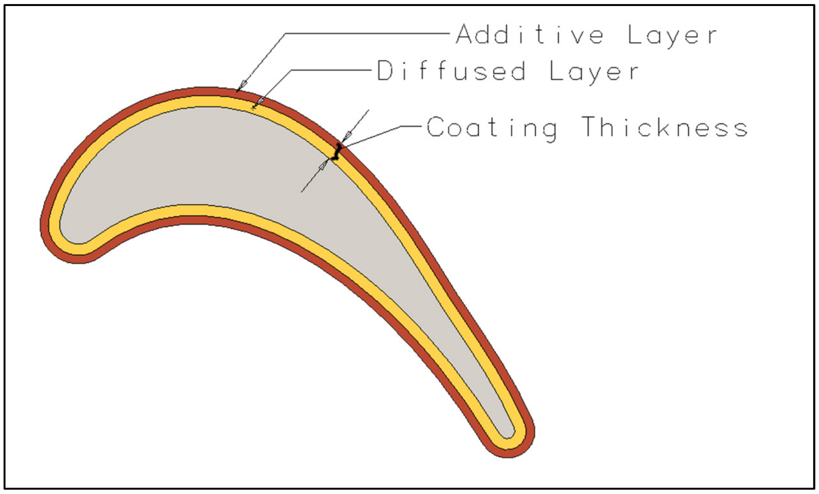


Figure 5-27 Representation of Coating Thicknesses

The total mass of the section is a summation of base metal mass and all the coating.

$$Total\ Mass = Mass_{metal} + Mass_{coating-1} + Mass_{coating-2} \tag{6}$$

This leads to the assessment of the 1-D stress (white line in ksi) as well as the mission (yellow in hours before predicted structural failure) and block test life (magenta in cycles before structural failure) as shown in Figure 5-28.

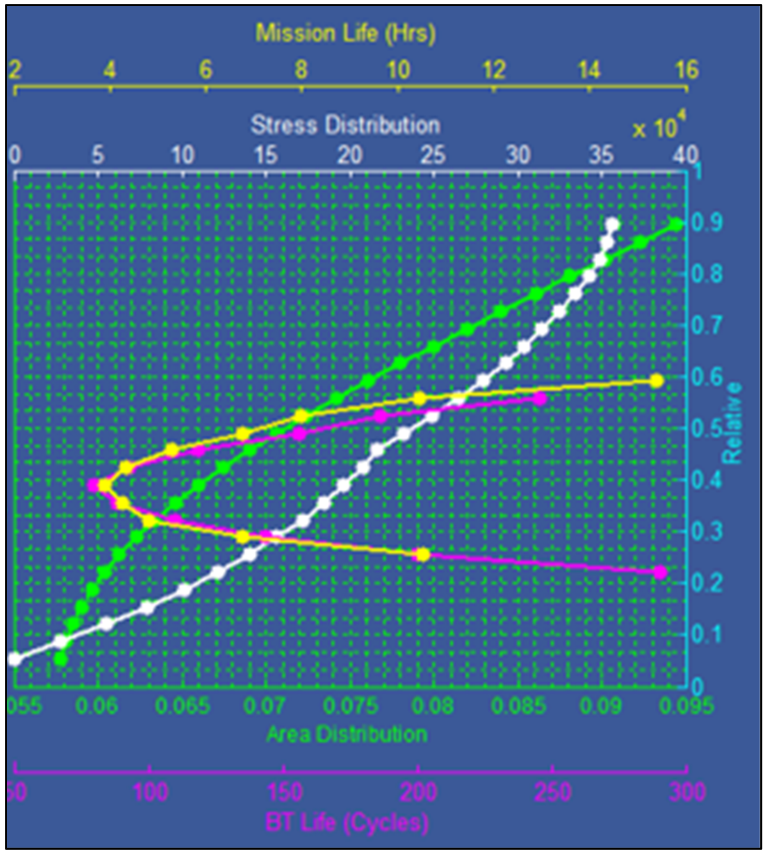


Figure 5-28 Example of Structural Results of Airfoils
(dummy airfoil)

5.6.2 Aerodynamic Analysis

The aerodynamic analysis of the airfoil is started in much the same way that the structural analysis is, cutting of the airfoil to prepare for analysis. During the pre-processing, the airfoil is meshed and the boundary conditions applied. An example of the mesh is shown in Figure 5-29. An O-mesh is applied around the airfoil to more adequately capture the boundary layer and an H-mesh captures the mainstream flow.

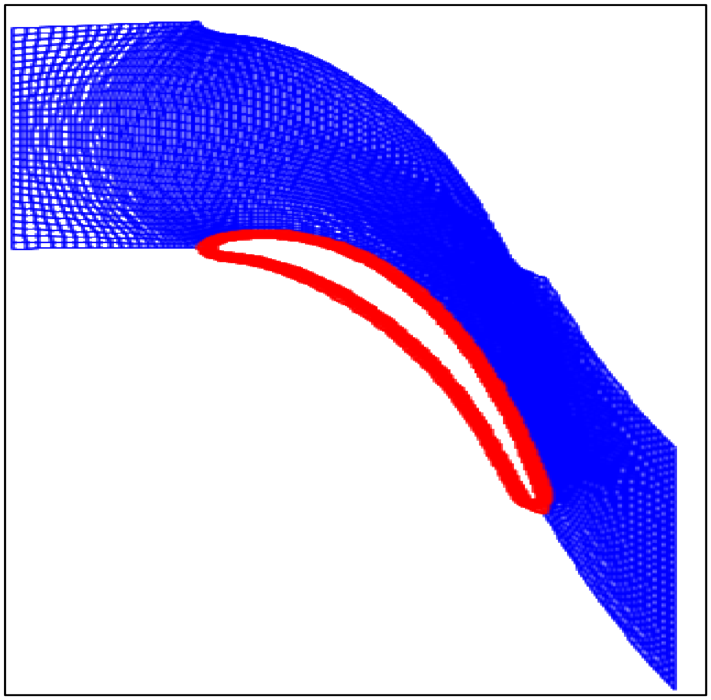


Figure 5-29 Example of Airfoil Mesh

The specific airfoil is then placed in the desired location in the turbine gas path, an example of which is shown Figure 5-30.

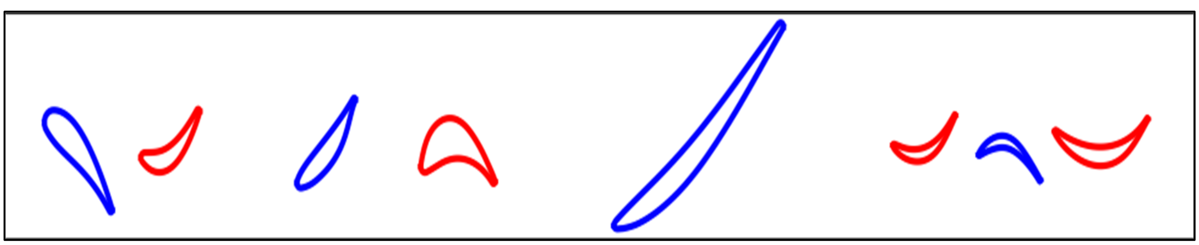


Figure 5-30 Example of a Turbine CFD Setup

The airfoil is then run using an in-house CFD solver (same one as that used in detailed design) and then post-processed to obtain aerodynamic parameters such as efficiency and stage reaction.

5.6.3 Surrogate Assisted Optimization (SAO)

To validate that the architecture does allow for the multi-disciplinary optimization (req. 15) an airfoil that was designed using traditional methodologies during pre-detailed design and the first few cycles of detailed design was designed using iRSO and optimized using surrogate assisted optimization (SAO). The SAO methodology was developed at P&WC with in a frame work for design experiments (FDE) that brings to bear the power of SAO with the flexibility that comes with in house codes. Papers have been written on how FDE was developed (Doran, et al., 2018), so the author shall only summarize the key points that are relevant to the work on iRSO. Whenever the author refers to SAO, he is referring to SAO as implemented in FDE.

As shown in Figure 5-31, SAO is started by reading a settings file that defines the SAO to be analyzed. This file has the constraints and objectives functions defined as well as a pointer to the process that is to be run. The program then builds a surrogate model and searches around where the last best point was found for a better point. To mitigate against the problems inherent to the gradient approach around the optimum (namely local maximums), the program also randomly selects a few points from the complete design space given. These set of points are given to the program (in this case iRSO) to run and the outputs assessed. The results are stored in a cloud database and the surrogate model updated with them. The SAO stops when a convergence criterion that the user sets (i.e., % change in results over subsequent runs) is reached or when a maximum number of runs that the user has specified is reached.

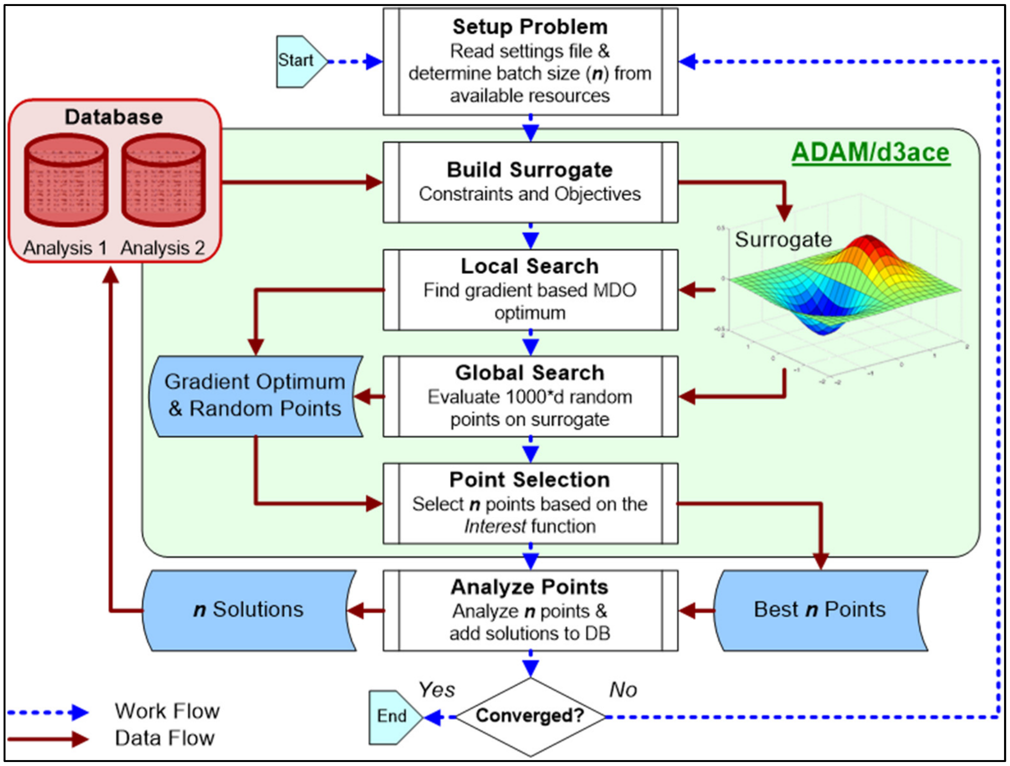


Figure 5-31 SAO Workflow (Doran, et al., 2018)

At this point, it worth remembering that the point of this thesis is not to show any one optimized rotor but rather to create an artifact that is optimizable. The mere fact that the author was able to setup and execute multiple optimization runs is proof that the objective has been achieved. None the less, engineering curiosity leads the author to review the results.

5.6.4 Results

The multi-disciplinary optimization of the airfoil results showed the true power of the iRSO. Optimization was launched with an initial an optimized geometry and only allowed to change the parameters that would not necessitate a new meanline analysis (axial chords, metal angles and the inner and outer gas path diameter) with the objectives of:

- increase efficiency
- decrease weight of the airfoil

And the constraints of keeping:

- the airfoil stress at around a certain span (P&WC best practice).
- peak stress below a certain percentage of material properties (P&WC best practice).
- reaction around the initial levels.

The results of the optimization were interesting to say the least. They confirmed things that were already well known and provided a deep insight into others. The results are shown in Figure 5-37 and Figure 5-38. Figure 5-32 to Figure 5-36 are a result of a study to gain deeper insight into the data by looking only at the variation of a single parameter and see its influence on an output parameter. This methodology allowed one to be able to see the most influential parameters to a certain key output.

From Figure 5-32, we can see that the most influential parameters that affects the weight of the airfoil is the throat openings, with the one at the hub being the most influential. This is to be expected as the smaller the throat opening is the more cambered the suction side is. With the axial chord not being a parameter the program could affect, the camber of the suction side becomes the parameter with the greatest influence of area at each section and hence the overall weight of the airfoil. The reason why the mid value is so much more powerful is also quite understandable. In a three section design the mid-section influences more of the airfoil (above and below the section) while the hub and tip would have less than half its strength as they influence only above it and below it respectively. Figure 5-34 also shows the same trend for the same reason as the total weight.

Figure 5-33 shows that the hub throat opening is the most influential when it comes to the maximum stress. This again makes sense as the maximum stress location in this specific airfoil is closer the hub. The larger the area around the hub the lower the stress and vice versa.

Figure 5-35 and Figure 5-36 show more of the aerodynamic aspect of the analysis. They show that the mid throat has the highest effect on stage reaction and intern stage reaction has the highest effect of efficiency. As the name implies, the size of the throat directly affects the flow that passes through the blade and therefore directly affects the stage reaction. For the same reasons as were highlighted in previous sections, it is understandable that the mid section has the greatest effect.

Figure 5-37 provides a deep insight on the aerodynamics of this blade. It shows two fascinating things:

- reaction has a strong influence in the efficiency (eta_overall_s3) of the blade, as shown in Figure 5-35 and Figure 5-37
- relationship between reaction and efficiency is parabolic one. An increase in reaction results in an increase in efficiency until at the maximum is reached then a decrease in efficiency follows

Figure 5-38 is the most interesting. In some cases, turbine design involves a compromise between structural requirements and aerodynamic requirements such as efficiency, this relationship is sometimes viewed as inversely proportional. This graph however shows that for this airfoil, the relationship is parabolic. At low weight values, the relationship is proportional but as the weight increases past the inflection point the relationship is inversely proportional. All this insight provides the engineer with a deeper understanding of the blade. One was able to ask the “what if” questions and, and as was the goal of the artifact, all were answered.

- If I increase the weight, what will happen to efficiency?
    - At low weight values, the relationship is proportional but as the weight increases past the inflection point the relationship is inversely proportional
- If I increase the stage reaction will the efficiency increase?
    - The reaction and efficiency relationship is a parabolic one. An increase in reaction results in an increase in efficiency until at the maximum is reached
- If I increase my hub area, how will my peak stress be affected?
    - The larger the area around the hub the lower the stress and vice versa.

All of these “what if” questions were answered, and that was the goal of the artifact.

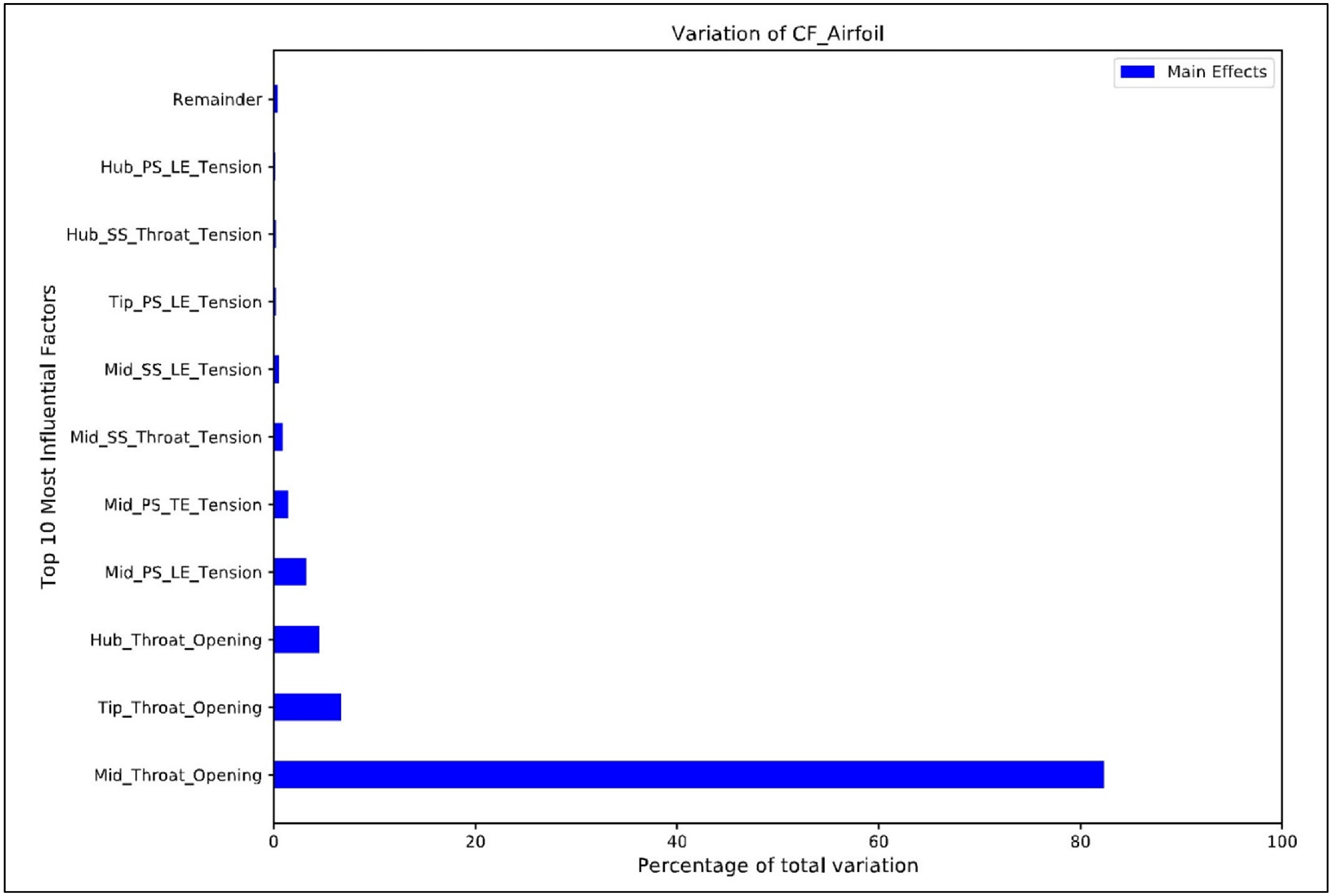


Figure 5-32 Airfoil Optimization Results - Weight Sensitivity

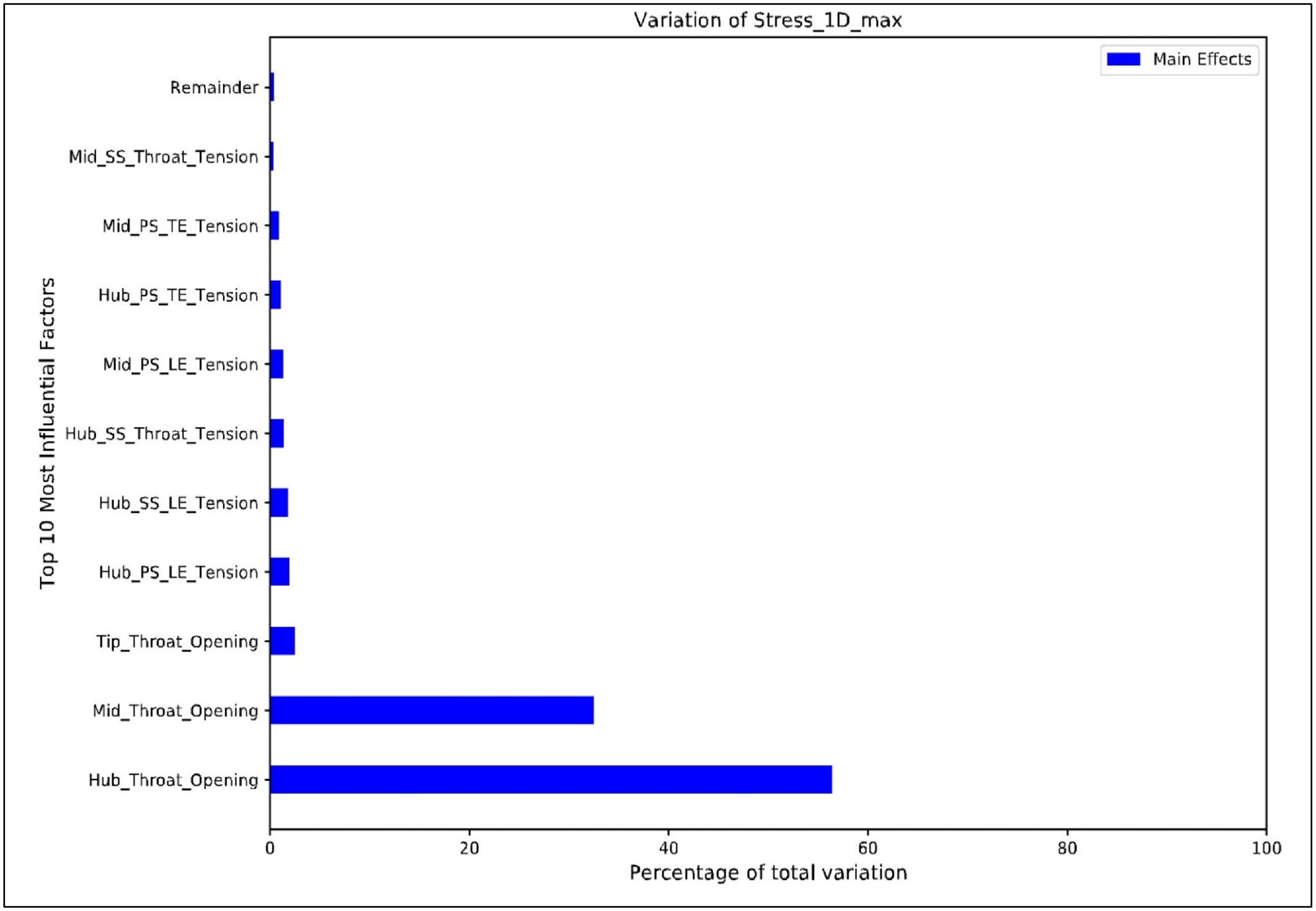


Figure 5-33 Airfoil Optimization Results - Max Stress Sensitivity

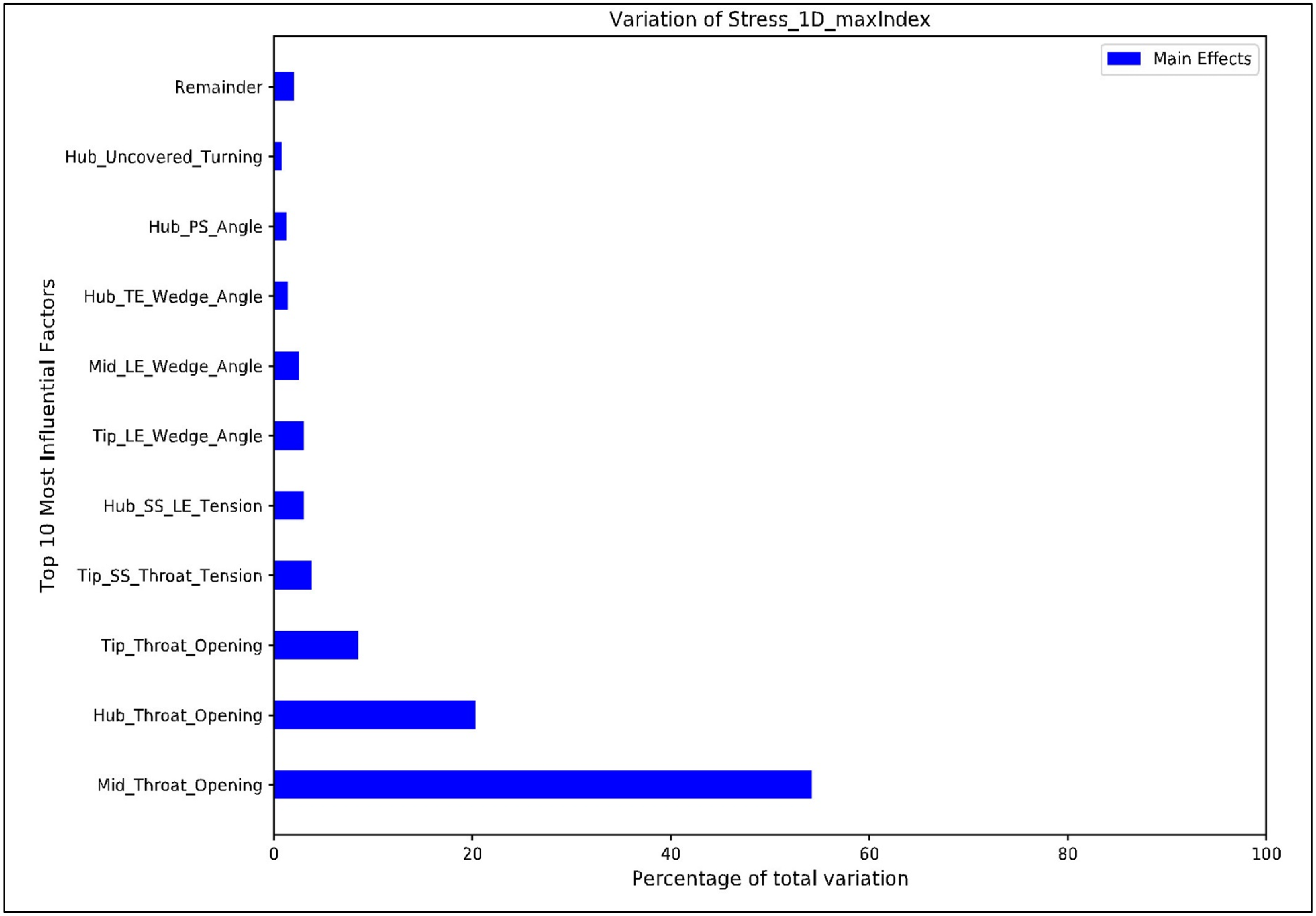


Figure 5-34 Airfoil Optimization Results - Max Stress Location Sensitivity

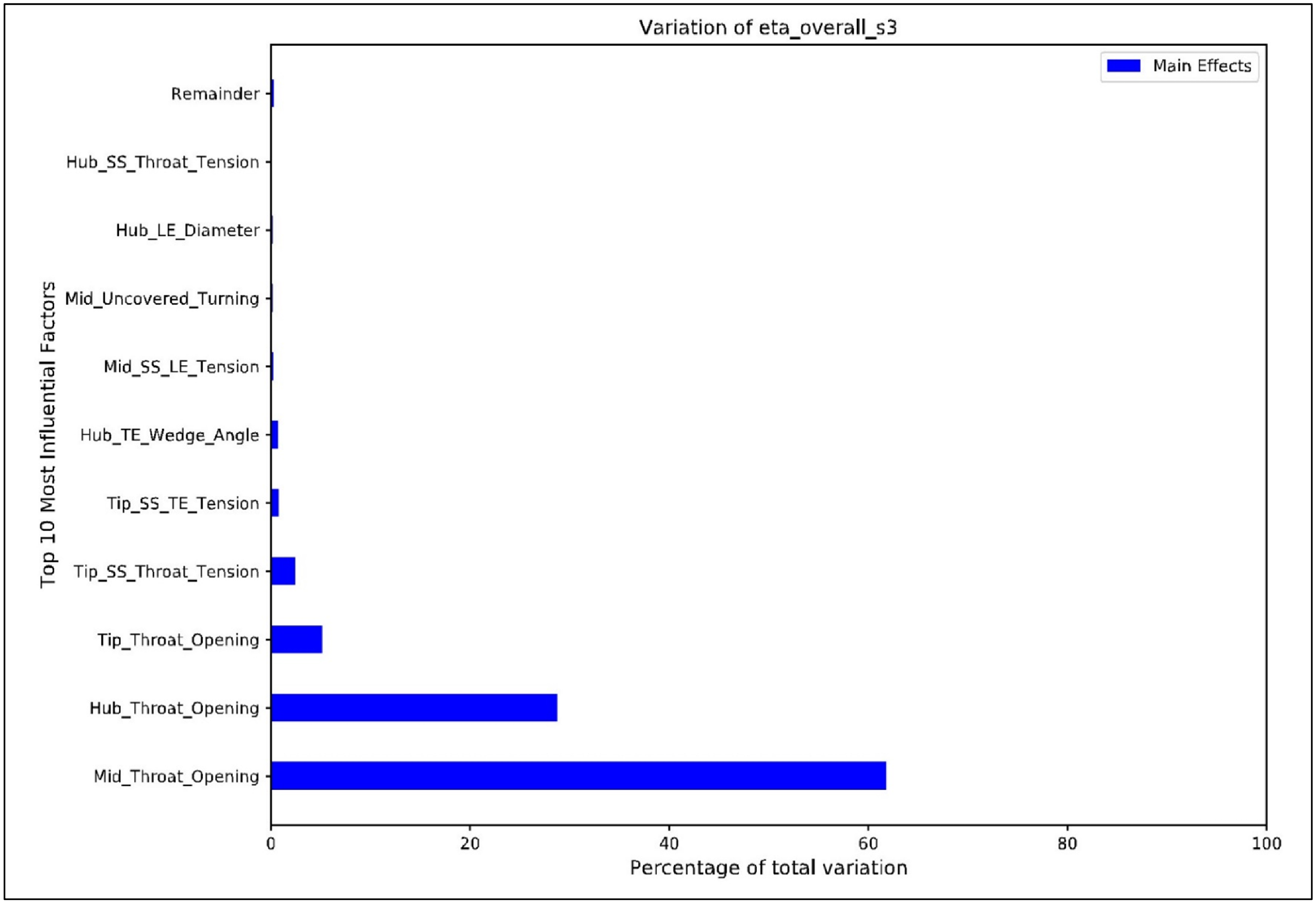


Figure 5-35 Airfoil Optimization Results – Efficiency Sensitivity

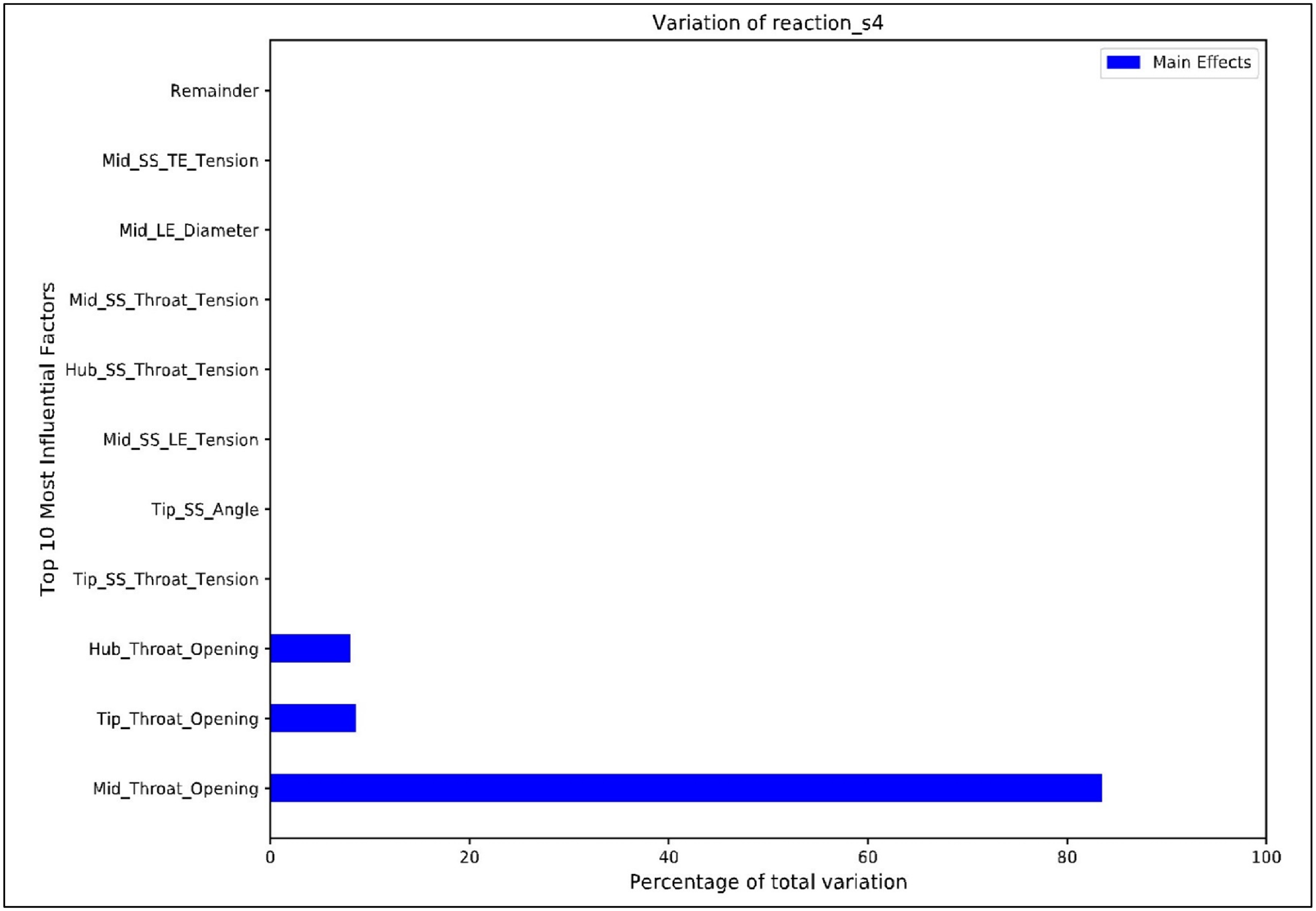


Figure 5-36 Airfoil Optimization Results – Reaction Sensitivity

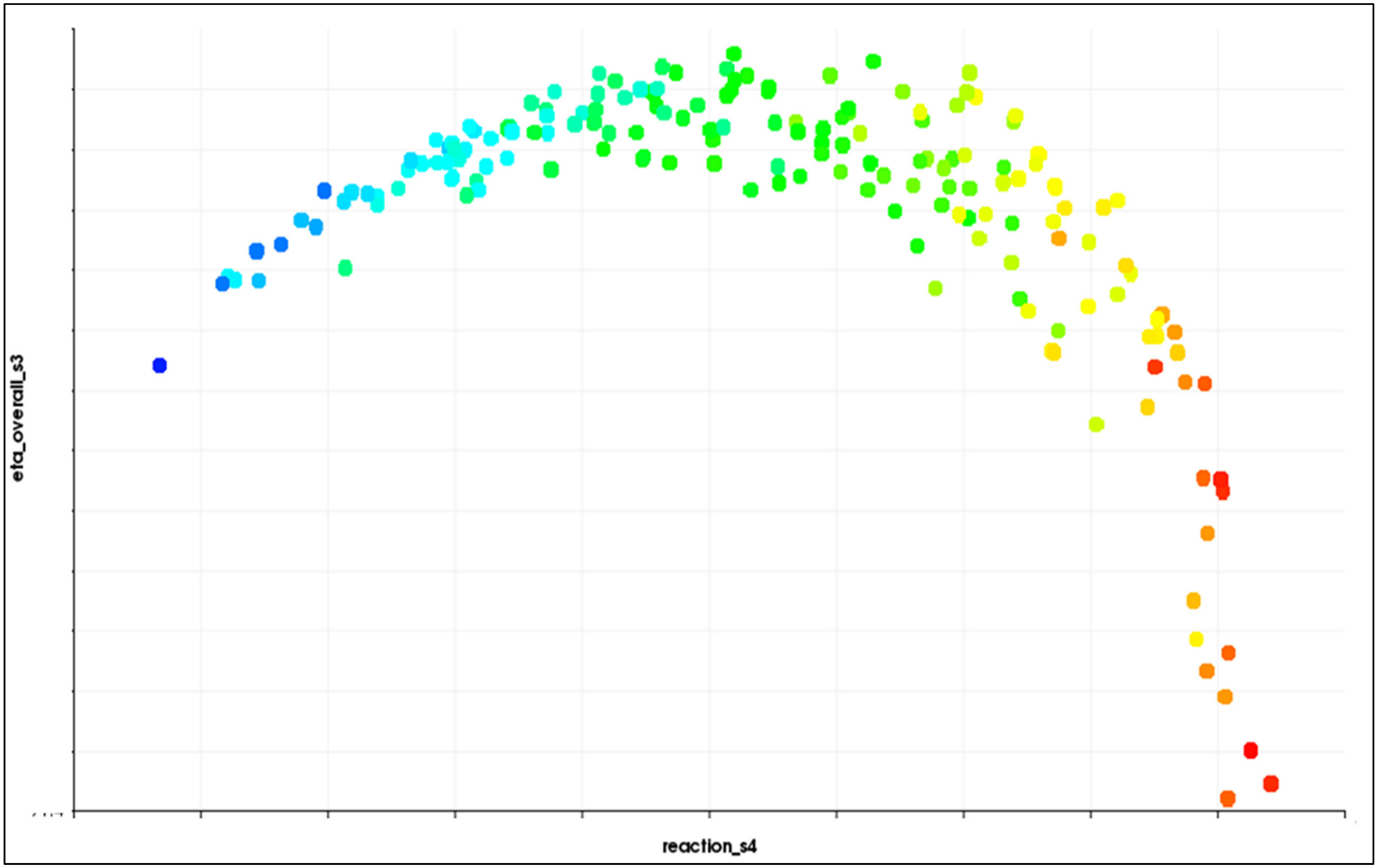


Figure 5-37 Airfoil Optimization Results - Efficiency vs Reaction

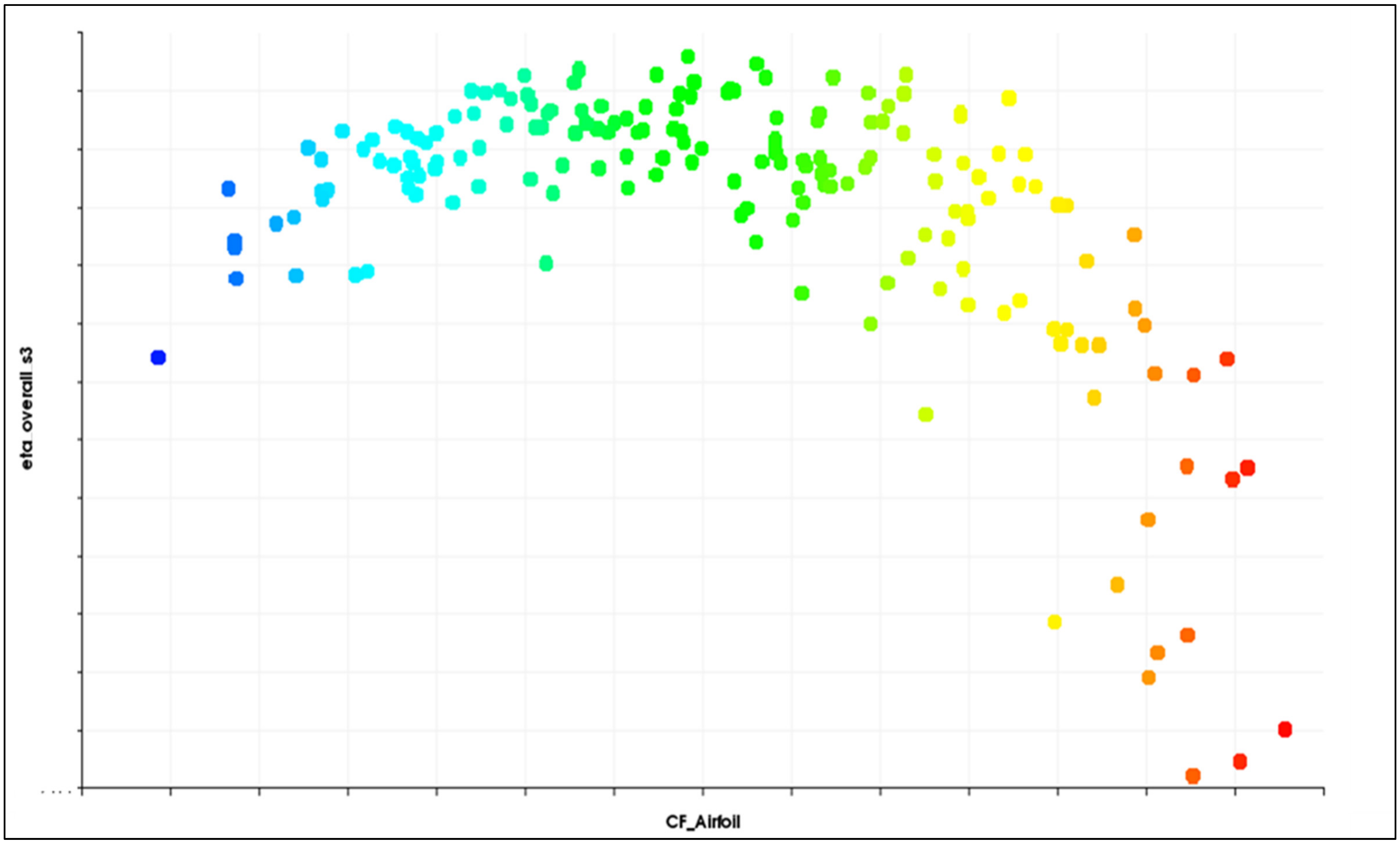


Figure 5-38 Airfoil Optimization Results - Efficiency vs Weight

Another example that shows the effectiveness of the artifact developed is in the work of Mavroudi (Mavroudi, et al., 2019). Using the fixing Design and Analysis (D&A) tool created by the author of this thesis within iRSO module that was able to design and analysis the structural integrity of the fixing (Twahir, et al., 2014). Mavroudi was able to run 1D analysis as well as quasi 3-D FEA analysis. The FEA analysis schematic is used in the work of Twahir et. al. (2014) shown in Figure 5-39

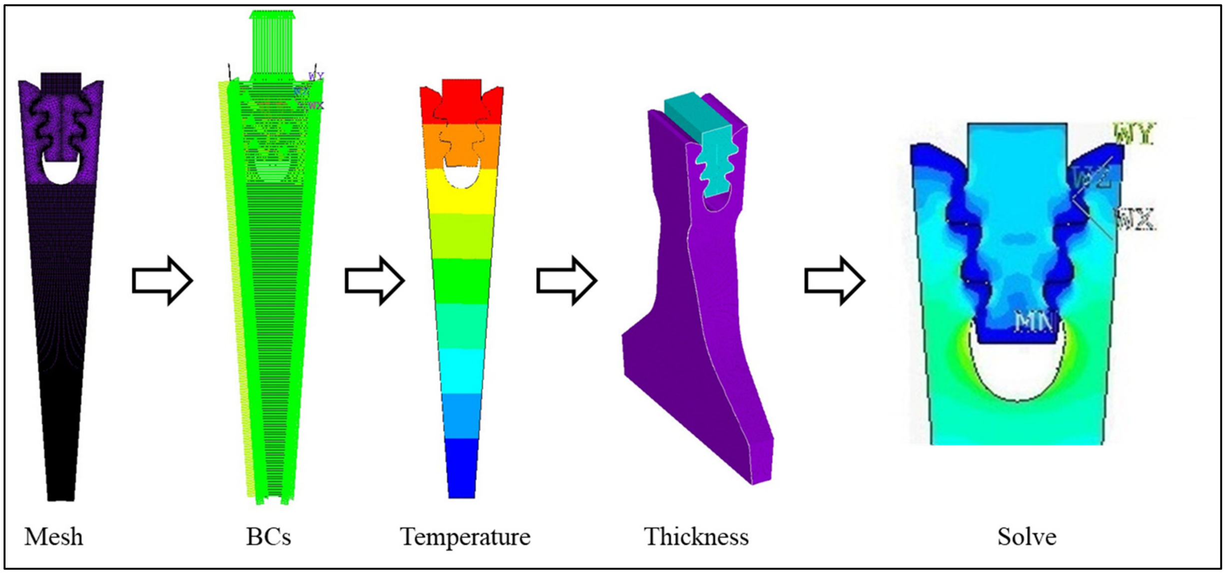


Figure 5-39 Automated Fixing FEA

Mavroudi's work concentrated on the optimization of the fixing. Her goal was to minimize the centrifugal load which would result in a reduction of the fixing weight and disc weight. She used SAO as was done for the airfoil in sub-chapter 5.6. She successfully ran the optimization for an uncooled three lobe fixing and a cooled two lobe fixing, both of which were to fit on the same blade. As was mentioned previously, the mere fact that she was able to run the optimization is proof that the architecture works but none the less engineering curiosity leads us to look at her rather impressive results. For the 3 lobe design, she was able to reduce the centrifugal load of the fixing by 33.3% from an initial fixing manually designed by an experienced engineer. The two lobe design showed a reduction of a more modest 12.96%.

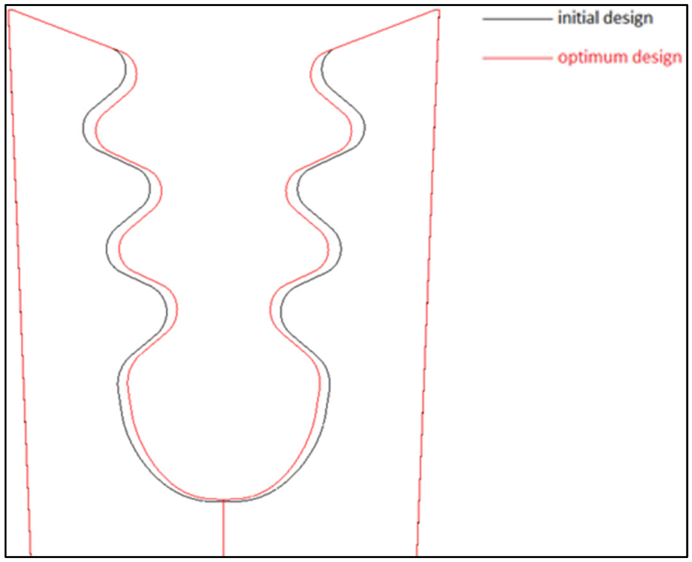


Figure 5-40 Disc Fixing Optimized Using iRSO – 3 Lobe (Mavroudi, et al., 2019)

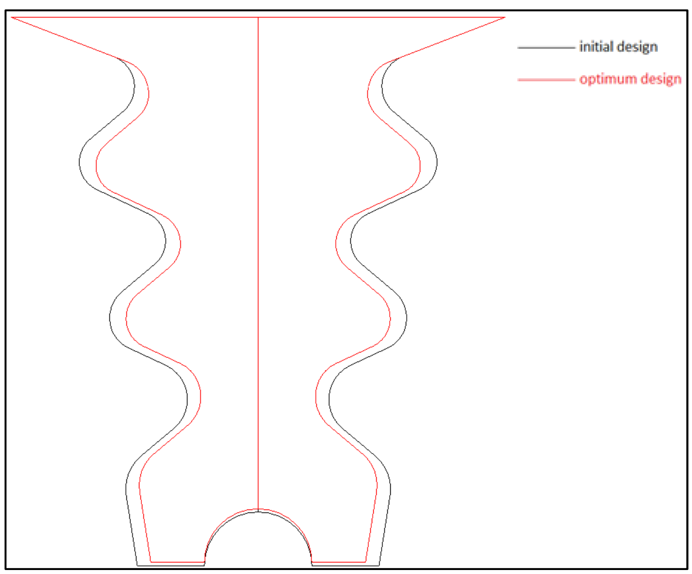


Figure 5-41 Blade Fixing Optimized Using iRSO – 3 Lobe (Mavroudi, et al., 2019)

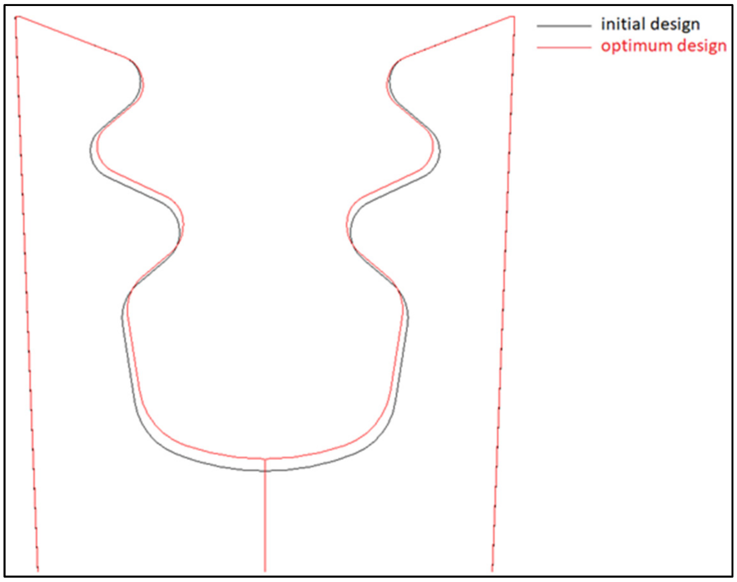


Figure 5-42 Disc Fixing Optimized Using iRSO – 2 Lobe (Mavroudi, et al., 2019)

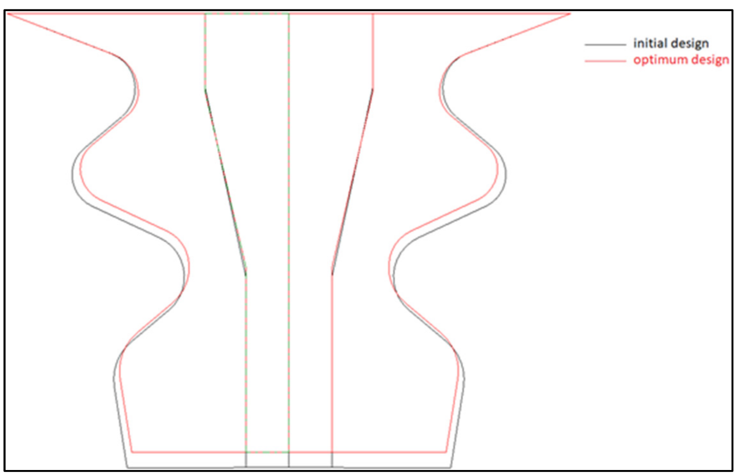


Figure 5-43 Blade Fixing Optimized Using iRSO – 2 Lobe (Mavroudi, et al., 2019)

"Without the developed optimization framework used in the pre-detailed design stage, the initial design of a fixing would mostly rely on established rules of thumbs" (Mavroudi, et al., 2019). These rules of thumbs with an experienced engineer might have provided similar results but not with the same amount of effort. The true utility is that it lowers the number of years of experience needed to design such a component as any engineer can run the artifact created (req. 6).

CHAPTER 6

FORMALIZE LEARNING

As mentioned before research in information systems / technologies (IS/IT) such as this must achieve dual objectives: to build knowledge or a theoretical contribution to the disciple as well as to assist in solving a real world problem in real world settings. At P&WC, a real need existed to create a system that could apply tenants of MDO at the pre-detailed stage (PMDO). The architecture created as well as the modules and tools developed using this architecture have increased the capacity of P&WC to explore many more design at the pre-detailed stage by reducing the time taken to design and increasing the fidelity.

This thesis presents theoretical contributions to the discipline through its novel:

- Methodological approach;
- Architecture;
- Modules and Tools;

Through the creation of an architecture for the design and optimization of gas turbine engine sub-assemblies and components, new methodological knowledge was created on how to best tackle a multidisciplinary design and create optimization capable tools and processes in an industry-educational institution collaboration. ADR as it was applied in this work may be used for any research. A new methodology in which power users are integrated into the research team, allowing them to be thoroughly engaged and gives the artifact a great advocate to the wider user base was also implemented.

An architecture was created that is scalable to a large number of users, extensible to allow for its future growth and for future functionalities. The architecture through the use of OOP is extremely modular enabling a high degree of re-use in both the classes and the functionalities of the classes. It also allows inexperience designer and programmers to develop good software therefore enabling people with various programming expertise to contribute effectively. Ease of maintainability is also achieved through the architecture and so is backwards compatibility.

Through the architecture multiple new modules and tools were created to facilitate the multi-disciplinary design and analysis of turbine rotors at the pre-detailed stage. The tools facilitate geometric representation through simple, robust and flexible CAD parameterization. Through access to multi-level fidelity analysis within the tools, iRSO was able to increase the level of definition of the turbine exiting the pre-detailed thus reducing the time taken in detailed design (around 30%) and reducing the risk associated with design leaving pre-detailed design. The graphical user interfaces created allowed for the injection of judgement and creativity, ensuring the PMDO was not a push button solution but rather one that keeps engineers in the center of engineering by helping them get more holistic answer to the various "what-if" questions they might have. The modules and tools created were easy to use with users needing minimal training before they could effectively use it. It also lowered the number of years of experience needed to design a component through the incorporation and automation of engineering standard processes and the incorporation of knowledge from more experienced users.

The close cooperation between the author as well as the PDDS team with the client through the power users that ensured frequent face-to-face conversations, coupled with the architecture (through its use of OOP) allowed for the rapid deployment of the artifact. This ensured the author was continuously delivering valuable software.

CONCLUSION

The pre-detailed stage of the design of the turbines in a gas turbine engine is critically important as it is when the technical envelope of the engine is set. This phase of design is however defined by lower fidelity tools that prioritize speed. This leads to large factors of safety having to be applied to any solution obtained from the lower fidelity tools and sometimes to lost opportunities due to fact that a response to an RFP may not as competitive as it could be on performance, life or cost. The opposite is also possible where the proposed response is too aggressive and large amount of warranty cost are paid as contractual obligations committed to after pre-detailed design are not met when detailed design is finished. Considerable amount of time is also lost in non-value added tasks of the copy-pasting and general conversion of an output of one program to an input of another. This again could result in lost opportunities as during the limited time allotted for pre-detailed design, fewer designs could be explored. This research was conducted to eliminate this uncertainty by deploying a fully automated and integrated turbine system capable of optimization. This would allow for knowledge to be brought forward in the design process to the pre-detailed phase where more design freedom is available, enabling P&WC to conduct engine development faster and reduce risk when it comes to responding to RFP from airframe OEMs.

This work created an architecture for the turbine design and optimization as well as a larger architecture for future work on other engine assemblies (compressors, combustors etc.). It also created a new process for the multi-disciplinary design and optimization of turbine rotors at the pre-detailed stage (iRSO) and a new tool for the system level thermodynamic parameters (EDS) needed during the design of the turbine rotors. Throughout the creation of these tools, ADR provided a methodological context for the solving of these practical problems. "Business needs motivate the development of validated artifacts" (Wieringa, 2009) that meet the needs of the stakeholders and generalize the solution to solve similar set of problems (Sein, et al., 2011).

As stated by Wieringa, to assess the solution to the practical problem one must "investigate how a proposed solution would achieve a goal in a given context" (Wieringa, 2009). This chapter explores how each of the objectives outlined in chapter 2.3 and 4.1 have been achieved by the work presented. The objectives outlined were to:

1. continuously provide value for P&WC.
2. reduce the pre-detailed time of engine development by a factor of 10.
3. reduce tools specific training needed to design a component; The training referred to here is training required to be a user of the tool and does not refer to the decade or so it normally takes an engineer to become skilled enough to understand the technical implications of what this tool will output as data.
4. facilitate engineers to control the design process and to inject judgement and creativity.
5. have flexibility in modeling and analysis that allows the reproduction of all current rotors design and creation of future design.
6. create a system that is scalable for future expansion.
7. lower the risk of the designs leaving concept design phase and pre-detailed design phase.

Through the rapid deployment of the tools and modules that were usable, we were able to ensure customer satisfaction by early and continuously delivering valuable software. This rapid and continuous deployment was made possible by the integration of a sub-set of the users (power users) into the research team, continuously getting useful feedback, shaping and reshaping reciprocally.

The incorporation of multi-level fidelity best in class tools such as 1-D calculation and commercial FEA and CFD software allowed for an increase of fidelity in the pre-detailed stage to a level usually found after a few cycles in detailed design. The high fidelity results would lead to an even more accurate response of RFPs leading a higher level of certainty on the ability of the engine to meet performance and life targets.

As mentioned before, the integrated system and its architecture allowed significant time savings on the pre-detailed phase but more impressively is the fact that the higher fidelity results allowed for the elimination of at least one design cycle in the detailed-design phase. This greatly reduced the total engine design time (pre-detailed, 80%, plus detailed design time, around 30%) and significantly reduced the cost outlay. A more in-depth business case study would be necessary to exactly quantify by how much the cost outlay is reduced by. However anecdotal evidence is sufficient for acceptability, deployment within the company and the development of future modules.

Through the creation of graphical user interfaces (GUIs) the user is able to see all inputs, outputs and the generated geometry all in one screen. This allows for the user to remain central to all actions of iRSO. This is in line with Sobieszczanski-Sobieski & Haftka observed that "[P]MDO, emphatically, should not be used as a push-button design procedure. The engineering design process moves forward by asking "what if" questions and using the answers to make design changes". "The human interface is crucially important to enable engineers to control the design process and to inject judgement and creativity" (Sobieszczanski-Sobieski & Haftka, 1997).

The GUIs also allowed for some of the intricacies of running an analysis to be simplified and only the parameters that truly affect the design could be concentrated on. The quick feedback and satisfaction provided by the ability to alter a parameter and to receive a CAD representation in the GUI as well as its structural or aerodynamic results within minutes allowed for the rapid build-up of experience hence lowering the number of years of experience needed to design a component. This also provided an unexpected benefit: fun! During the design of an actual engine component, the feedback the researchers got was that the engineers were having fun with the tool.

Underpinning the GUI and all the tools was a good representation of the product. This was achieved by the parameterization of all required turbine parts. The parameterized CAD objects were able to model all the different configurations currently at P&WC effectively and efficiently. The greatest compliment that was paid to the parameterized models created for iRSO is that the same models were used in the early stages of detailed design.

Through the work done by Mavroudi (Mavroudi, et al., 2019), it was shown that the system created was optimization capable. The centrifugal load of the fixing was reduced by 33.3% thus reducing the overall weight of the disc and the rotor in general. Her work used both the 1D calculation for the fixing as well as the 2D data from FEA. The calibration of the 1D data from the 2D allowed for quicker convergence to an optimized solution.

The multi-disciplinary optimization of the airfoil (combining structural and aerodynamic requirements) showed that the optimization algorithm was able to reduce the weight of the airfoil by around 10% and increase the efficiency by 0.3%. Although these results were not shown to have led to an airfoil that was ready for casting, it did give a window into the eventual capability of the system.

All in all, the architecture created allows for a system that is scalable for future expansion. The architecture created using object-oriented programming (OOP) is exhaustive in outlining all major parts of the engine. On the turbine assembly, the structure was detailed and allowed for the modeling of all components (shroud, airfoil, platform, fixing, disc, and cover-plate). It was also found to be extensible, allowing for the future addition of the compressor.

The architecture also created only one copy of any particular parameter but allowed all parameters to be accessible from all parts of the structure. This stopped possible mistakes as all tools worked with the same data and did not make copies of data. It also eliminated the non-value added tasks of transferring of data from one tool to the other, a big part of the more than 80% reduction in time taken when compared to previous pre-detailed design process.

The architecture also modularized certain aspects of the analysis such as the FEA and CFD. By implementing dedicated interfaces with the commercial software, any future engineer would need to change the code in one location to affect how any turbine component is analyzed, allowing for the easy maintenance of the system.

This thesis showed how ADR could be used in the industry in partnership with academia with great results. The adherence to the stages and principles of ADR created an artifact that accomplished all objectives set for it. It managed to assist in solving a real world problem in real world settings as by reducing the amount of time taken for the pre-detailed stage of design but it also showed the ability to eliminate a highly expensive cycle of detailed design, thus saving the company money. It also built knowledge and made a theoretical contribution to the disciple through the creation of a new methodological approach and architecture.

RECOMMENDATIONS

With the successful completion of this thesis work, the power of distance and time gives a perspective of what could be improved on and changed to create a tool that gives even more utility to P&WC.

1. An organizational cultural shift must occur to fully harness the potential of iRSO. Instead of thinking of one's self as belonging to a specific discipline, generalists are needed who are well versed in all the disciplines related to turbine rotor design to use the full power of iRSO.

2. More work should be done to create component optimization flows. A lot on insight can be gained from the data that can be produced. What would have taken months or years of experience can be gained through the proper study of the sensitivities that are a result of SAO as shown in paragraph 5.6.

3. With every tool, the continuous utility is gained through great maintenance. The resulting code is more than 500,000 lines long. A good understanding of how it works and how to maintain it is needed.

4. Finally, in order to reach the full potential of the PMDO system developed by the combined initiative of Pratt & Whitney Canada and the École de technologies supérieure, the set of tools produced during this Ph.D. study should be used along with all the other required tools to run optimisation loops on the whole turbine design and analysis process.

LIST OF BIBLIOGRAPHICAL REFERENCES

Brophy, F., Mah, S. & Turcotte, J., 2010. *Preliminary Multi-Disciplinary Optimization (PMDO) an Example at Engine Level.* Berlin, NATO Science and Technology Organization.

Doran, P., Vlasic, E., Guevremont, G. & Moustapha, H., 2018. *Gas-path Optimization Using Turbine Aerodynamic Meanline and Design Exploration.* Montreal, Global Power and Propulsion Society.

Dulikravich, G., 1991. *Aerodynamic Shape Design and optimization.* Reno, NV., American Institue of Aeronautics and Astronautics Inc.

Goel, S., 2009. Turbine Airfoil Optimization Using Quasi-3D Analysis Codes. *International Journal of Aerospace Engineering,* Volume 2009.

Groh, R., 2013. *Aerospace Engineering Blog.* [Online] Available at: http://2.bp.blogspot.com/-_WUOXSjMAq8/Tw1oj9VtXOI/AAAAAAAABkE/CprbcSy0S18/s1600/31.JPG [Accessed 05 December 2019].

Hutchison, M. et al., 1994. Variable-complexity aerodynamic optimization of a high-speed civic transportation wing. *Journal of Aircraft,* pp. 110-116.

Jha, R., 1999. Development of Multidisciplinary Design Optimization for Smart Composit Wings and Turbomachinery Blades. *Dissertation Abstracts International.*

Korte, J. J., Weston, R. P. & Zang, T. A., 1998. *Multidisciplinary Optimization Methods for Preliminary Design,* s.l.: Multidisciplinary Optimization Branch, MS 159, NASA Langley Research Center.

Lagloire, F., Oullet, Y., Blondin, B. & Moustapha, H., 2013. *Single Platform Integration Environment for Turbine Rotor Design & Analysis.* Busan, Korea, American Institute of Aeronautics and Astronautics.

Longo, A., No, M., Aizpitarte, M. & Unzueta, J., 1991. Conservatice approximations in non linear optimizaion; theory and examples. *Computer & Structures,* pp. 441-449.

Martins, J. R. R. A. & Lambe, A. B., 2013. Multidisciplinary Design Optimization: A Survey of Architectures. *AIAA Journal,* Vol. 51(No. 9), pp. 2049-2075.

Mavroudi, D., Abenhaim, A.-I., Moustapha, H. & Kalfas, A., 2019. *Design Optimization of Aeroengine Turbine Blade and Disc Fixing.* Montreal, CASI Aero Conference.

Moradi, N., Vlasic, E. & Moustapha, H., 2015. *Rapid Airfoil Design for Uncooled High Pressure Turbine Blades.* Montreal, ASME.

Moret, M. et al., 2017. *Propulsion System Integration and Optimization at the Preliminary Design Phase.* Cairo, Egypt, Aerospace Science & Aviation Technology.

NATO RTO Research Task Group AVT 093, 2006. *Integration of Tools and Processes for Affordable Vehicles,* Neuilly-sur-Seine Cedex, France: NATO RESEARCH AND TECHNOLOGY ORGANISATION.

NATO Science and Technology Organization, 2006. *Integration of Tools and Processes for Affordable Vehicles. Chapter 3: Air Vehicules,* s.l.: NATO RTO Research Task Group AVT 093.

Ouellet, Y., Garnier, F., Roy, F. & Moustapha, H., 2014. *A Preliminary Design System for Turbine Discs.* Düsseldorf, Germany, ASME.

Ouellet, Y. et al., 2016. A Preliminary Design System for Turbine Discs. *International Journal of Turbo & Engine Jet,* 36(3), pp. 329-338.

Panchenko, Y. et al., 2002. *Preliminary Multi-Disciplinary Optimization in Turbomachinery Design.* Paris, France: RTO-MP-089, s.n., p. 22.

Panchenko, Y. et al., 2002. *Preliminary Multi-Disciplinary Optimization in Turbomachinery Design..* Paris, France, s.n., p. 22.

Pandey, V. S., Lee, C.-P. & Wadia, A. R., 2012. *Shrouded Turbine Blade With Contoured Platform and Axial Dovetail,* European Patent EP2423438: General Electric Company.

Pratt & Whitney Canada, 2019. *Fast Facts.* [Online] Available at: https://www.pwc.ca/en/company/about-pratt-and-whitney-canada/fast-facts [Accessed 06 July 2019].

Rolls Royce, 2015. *The Jet Engine.* 5, illustrated, reprint ed. s.l.:John Wiley & Sons.

Saric, I., Repcic, D. & Muminovic, A., 1996. *Parameter Modelling of Gears.* s.l., Proceedings of the 14th International Research/Expert Conference „Trends in the Development of Machinery and Associated Technology.

Sein, M. et al., 2011. Action Design Research. *MIS Quartely,* 35(1), pp. 37-56.

Sobieszczanski-Sobieski, J. & Haftka, R. T., 1997. Multidisciplinary aerospace design optimization: survey of recent developments. *Structural optimization 14.1*, pp. 1-23.

Sobieszczanski-Sobieski, J. & Haftka, R. T., 1997. Multidisciplinary aerospace design optimization: survey of recent developments. *Structural optimization 14.1 ,* pp. 1-23.

Sullivan, K. H. & Milberry, L., 1989. *Power: The Pratt & Whitney Canada Story (Volume 1).* s.l.:CANAV Books.

Tayla, S., Rajadas, J. & Chattopadhyay, A., 2000. Multidisciplinary Optimization for Gas Turbnie Airfoil Design. *Inverse Problems in Engineering,* pp. 283-308.

Tayla, S. S., Chattopadhyay, A. & Rajadas, J. N., 2002. Multidisciplinary Design Optimization Procedure for Improved Design of a Cooled Gas Turbine Blade. *Engineering Optimization*, Volume 34(2), pp. 175-194.

Twahir, A., 2013. Preliminary Design of Blade and Disc Fixing for Aerospace Application Using Multi-Disciplinary Approach. *Dissertations and Theses,* p. 141.

Twahir, A., Roy, F., Moustapha, H. & Attia, M., 2014. *Preliminary Design and Analysis Tool For Aeroengine Turbine Fixings.* Montréal, ASME.

Wieringa, R., 2009. *Design Science as Nested Problem Solving.* New York City, ACM, pp. 1-12.