

Abstract

Designing a complex product requires many disciplines. Major technical difficulties associated with this are the creation of robust parametric geometry so that accurate analysis and optimization can be accomplished, and complexities related with software integration. The research in this paper demonstrates that commercial computer-aided engineering software has the ability to implement effective multi-disciplinary optimization. A design tool written in C++ using the Unigraphics® application program interface was developed for modeling of blade outer air seals, a jet engine component, to quickly create a robust parametric model. Next, separate analyses were integrated and automated within ANSYS®. Finally, the robust parametric model and automated analysis were combined to form an integrated design loop using i-Sight FD®. This provides engineers the ability to run a variance based sensitivity analysis or optimization to produce a commercially acceptable product.

1 Introduction

Designing complex products involves several disciplines. For example, the design of a jet engine includes, among others, disciplines such as aerodynamics, structures, thermodynamics, and manufacturing. Traditionally, individual departments within a company would handle a single discipline. As a result, these groups became experts at analyzing and optimizing the product in terms of their specific perspective. This product development structure has an inherent problem. Instead of optimizing a part with respect to all disciplines based on their relative importance, the part is optimized by each discipline separately and a compromise between the optimized designs is selected. Clearly, a new method of product development is needed.

An improved approach to product development is concurrent engineering. This method moves away from sequential design, which uses specialized departments, and instead emphasizes multi-disciplinary design using Integrated Product Design (IPD) teams (Bartholomew, 1998). IPD teams can implement Multi-Disciplinary Optimization (MDO) to find an optimal design and control the complexity of their coupled systems. Currently, MDO is not extensively used in industry.

There are two main challenges to performing MDO, computational expense and organizational complexity (Sobieszanski-Sobieski and Haftka, 1997). Each discipline has specific analysis codes and design variables. Integration of many design and analysis packages increases computational time. Furthermore, generating a single optimization routine is a significant challenge. There are a couple of specific obstacles and questions for the implementation of MDO that will be addressed in this paper. The American Institute of Aeronautics and Astronautics (AIAA) Multi-Disciplinary Design Optimization Technical Committee defined many of these questions (Giesing, 1998):

- CAD and Parametrics

Most companies use commercial CAD systems to define their product geometry. Are these systems and the parametrics used to define the

models robust enough to be used in an optimization loop (Rohl, et al., 1998)?

- Software Integration

There are several different disciplines involved. How are their analysis codes integrated together? In addition, how are different software packages integrated together?

There have been several software tools developed that aid in analyzing products. Some of these are Computer Aided Design (CAD), Finite Element Analysis (FEA), and Computational Fluid Dynamics (CFD) tools. Using these Computer Aided Tools (CAx) has enabled companies to achieve increased product quality, reduced cost, and time-to-market (Srinivsan, et al., 2001). The combination of these CAx tools, optimization software, and more powerful computers has lead to improvements in the design process. However, optimization has often been limited to specific disciplines with limited interaction between the disciplines (Townsend, et al., 1998).

This paper aims to demonstrate the ability of commercial software to successfully implement MDO. This approach is applied to a Blade Outer Air Seal (BOAS) of a jet engine. A BOAS is located in the turbine section of a jet engine directly above the turbine blade. Hot air flows past the turbine blade and the BOAS prevents the hot air from leaking out. The location of the BOAS within an engine is shown in figure 1 and a BOAS is shown in figure 2.

A parametric model is generated in Unigraphics[®] using a custom design tool. Next, an analysis tool in ANSYS[®] automatically runs a thermal and structural analysis. Finally, i-Sight FD[®] couples the design and analysis tool into an automation loop. The development of this loop using commercial software demonstrates that MDO can be implemented in industry.

This paper is organized as follows. In the next section, a literature review is provided. Section III discusses the development of the design while section IV explains the importance of an analysis tool for MDO. Next, section V gives the details of combining the design and analysis tools within i-Sight FD[®] to form an automation loop. Finally, conclusions and recommendations are provided.

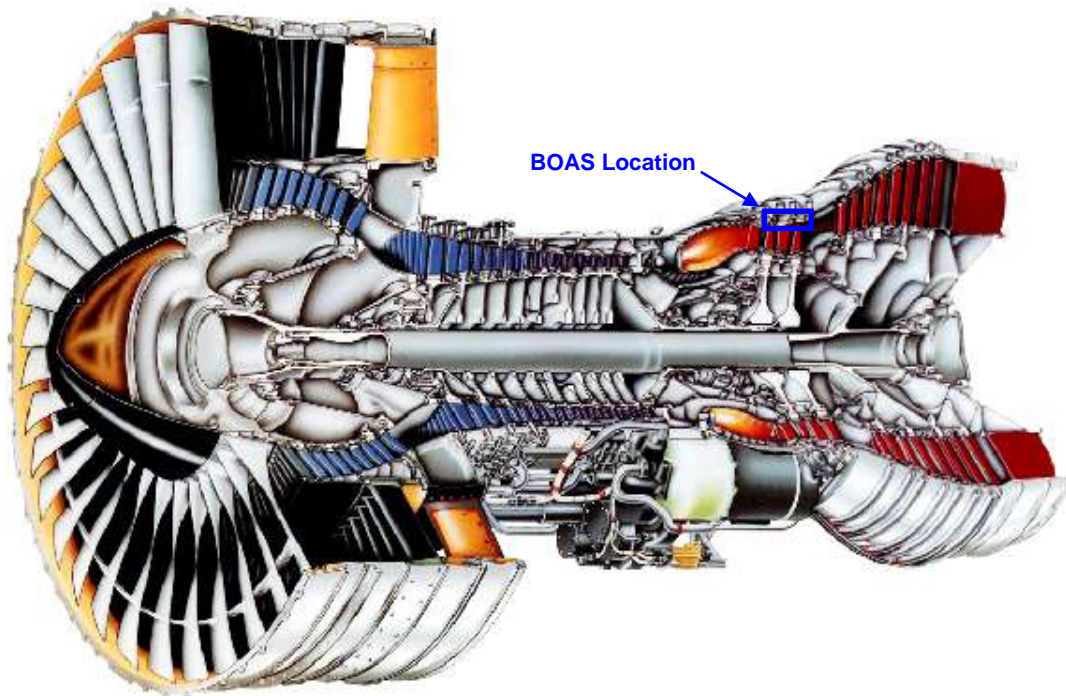


Figure 1: Jet Engine Cross Section (United Technologies- Pratt & Whitney)

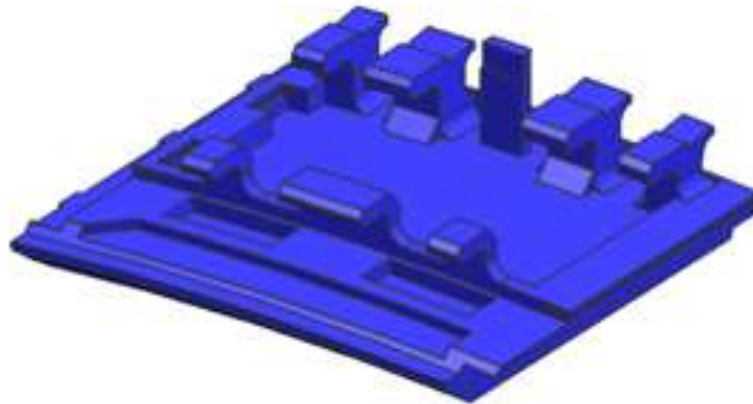


Figure 2: Blade Outer Air Seal (BOAS)

2 Literature Review

This section will provide a review of previous work on this subject. The following topics will be discussed:

- Concurrent Engineering
- Computer Aided Design (CAD) and Parametrics
- Multi-Disciplinary Optimization (MDO)

2.1 Concurrent Engineering

Concurrent engineering is a systematic approach to the integrated design and analysis of products. From the beginning, all disciplines work together on the design of a product.

This replaces the departmentalization scheme where each discipline would finish the design of the part from their point of view. Afterwards, they turn the design over to the next discipline. This creates a sequential design scheme. MDO can be effective in a concurrent engineering environment. With MDO, all disciplines are involved in all stages of the design process.

Over the past couple of decades, there has been great advancement in product design and development (Prasad, 1993). The product development cycle has been reduced by tools such as CAD, Computer Aided Engineering (CAE), Computer Aided Manufacturing (CAM), and Computer Aided Process Planning (CAPP). This has led to a more controlled manufacturing process (Prasad, 1994; Prasad, 1995). Furthermore, a feasible approach has been developed to integrate CAD and CAPP (Zhou et al., 2007).

This research will focus on integrated CAD, FEA, and optimization software. There is a need to integrate computer aided tools to improve computer engineering ability (Sevenler et al., 1993). If computer tools are used properly, they can achieve production without the preparation of full engineering drawings (Bralla, 1996).

Robust parametric tools, such as the design tool developed as part of this research, make the implementation of concurrent engineering easier. They provide a common model that all disciplines can work with simultaneously. This reduces the time it takes to design and analyze a part. Thus, a company can begin to produce a product quicker.

2.2 Computer Aided Design (CAD) and Parametrics

Parametrics has transformed the CAD industry. Parametric Technology Corporation's Pro/Engineer[®] introduced parametrics and it is now implemented by all major CAD packages (Hoffman and Kim, 2001). Parametrics defines the geometry parameters, relationships, and constraints. The geometry is morphed by varying these parameters. Thus, the model is changed and updated quicker than if the geometry were to be redrawn. Parametric models have been developed for jet engine turbine airfoils (Hogge, 2002). This demonstrated how parametrics help implement MDO. Parametrics can be implemented on several CAD and analysis packages (Jones, 2001). In fact, it has been shown that parametric information can be translated between different CAD systems with a programmatic approach (Tucker, 2000).

Storing product information as attributes is particular area where CAD research has been focused. Engineering knowledge and design intent can be captured through parameters (Ault, 1999). Information that is needed for downstream processes can be stored within the geometry (Anderl and Mendgen, 1995). New research has been conducted to maintain the meaning of the features during all modeling operations (Biddara and Bronsvort, 2000). Storing attributes within the geometry is helpful and acts as clues and hints for feature manipulation, specifically feature recognition and transformation (Subrahmanyam et al., 1995).

It is important to maintain the robustness of a parametric model. If the model is not robust, users of the geometric model and the optimization software will create nonmeshable geometry that cannot be interpreted by FEA software. This wastes the time of either the user or the optimization software. There are several factors that may cause a CAD model to fail such as poor coding, bugs in the CAD software, and poor parameterization. However, in most cases, the failure to regenerate a parametric model is due to the parametrization scheme

used to define it (Hoffman and Kim, 2001). Fortunately, it has been shown that robust parametric models can be created and used in optimization (Hardee et al., 1999).

Another area of research within CAD is design by features. The CAM-I project proposed and investigated feature-based design (Pratt and Wilson, 1987). In the mid 1980s, feature-based modeling systems appeared in university research labs. Some of these systems were at MIT (Miner, 1985), University of Massachusetts (Cunningham and Dixon, 1988), Stanford (Cutkosdy et al., 1988), and Purdue (Turner and Anderson, 1988). The research presented in this paper builds upon traditional feature-based design and utilizes user defined features to create BOAS geometry.

2.3 Multi-Disciplinary Optimization (MDO)

Multi-Disciplinary Optimization is being implemented to improve the design and analysis of complex products. Originally, MDO was applied in the aerospace industry. Codes from several disciplines were integrated to perform an optimization on the structural performance of a jet engine (Chamis, 1999). MDO has also been applied on other aerospace components (Tappeta et al., 1999). Other industries such as automotive and construction are starting to implement MDO. Developments and the current status of MDO are well documented (Sobieszanski-Sobieski and Haftka, 1997; Bartholomew, 1998; Lewis and Mistree, 1998).

3 Design Tool with Unigraphics®

There are several considerations that determine how a design tool is developed. Creating the correct design tool for the right situation ensures the tool will be useful.

3.1 Goals of a Design Tool and Intended Users

A customized engineering design tool needs to achieve several goals that are not easily attained. First, design time needs to be reduced. The engineer should spend less time interacting with CAD software when the design tool is implemented. Next, the design tool must produce geometry that is analysis capable. Analysis capable means the geometry can easily be meshed and interpreted by FEA software. Furthermore, the geometry needs to be parametric and robust. Specifically, engineers must be able to modify the geometry by changing the defining parameters without breaking the analysis capability. Finally, the design tool must be able to adapt to future designs and produce current designs. If it cannot, the design tool will be replaced with another method in the future. Achieving these goals results in engineers creating higher quality part models faster than before. This allows the engineers to spend more time on other engineering tasks.

3.2 Intended Users

When creating a design tool for engineers, the experience level of the end user must be considered. The type of end user determines the tradeoff between flexibility and automation. Typically, new engineers prefer to have a high level of automation because they are not familiar with the design process and the finer details of the part. A new engineer will only be aware of the main design features until they gain more experience. A high level of automation makes the tool easy to use and can quickly be learned. Unfortunately, the design

tool will not be as powerful since the end user would have fewer options available to them. The increased automation level decreases the amount of options.

In contrast, experts prefer to have a lot of flexibility since they want to attempt more complicated designs. Experts are familiar with the product and the design process. Therefore, they will be able to make more sophisticated changes to the design. This allows for the user to attempt many different design configurations and modify small details of the design. However, this makes it difficult for engineers to learn and implement the design tool. Furthermore, a flexible design tool is harder to create and it is difficult to make robust. Finding the right balance between automation and flexibility is paramount when creating a successful design tool.

The design tool described in the next sections is designed for experts. Thus, there is a high level of flexibility built into the tool. When new engineers are hired and expected to use the tool, the experienced engineers will have to train them. Furthermore, significant testing is required to ensure that the tool is robust since it is flexible.

3.3 Development Approach

There are two main approaches to consider when creating a design tool, the regenerative approach and the feature library method. Both have advantages and disadvantages.

3.3.1 Regenerative Approach

The regenerative approach reconstructs the entire geometry from scratch with each change. This is typically done with a set of input values that are read by a program that creates the geometry. For example, a user specifies geometric dimensions for a part and the design tool generates the entire geometry based on those inputs. To create a different design, the user gives a different set of dimensions and a new design is created.

The regenerative approach has many positive aspects. This method is appropriate for parts with complex geometry. The complex geometry can be completely handled within a program. This makes it robust because once the program is fully debugged; it will not cause an error. Furthermore, this method is easy for the user. The engineer only needs to enter geometric parameter values and run the program. Therefore, the user does not need to be an expert at using the CAD software.

The regenerative method is not without its flaws however. First, this approach has difficulty with topology changes. Each small topology has to be able to exist with all other topologies that are logical for it to work with. There can be too many possibilities for the programmer to handle robustly. Another problem is that users cannot improve the design tool unless they are a programmer. This method requires that a programmer write and maintain the design tool code. This is a problem when the program needs to grow and adapt to new designs and technologies and there are only a select number of people that can improve the design tool. Finally, the last drawback is that the entire geometry needs to be recreated by the program to make the smallest change. This can be computationally time consuming for complex geometries. The regenerative approach is certainly applicable to some but not all types of geometry.

An example of a part that would benefit from the regenerative approach is the turbine blade shown in figure 3. The turbine blade shown has a complex configuration. The core, which is not shown, has a complicated shape and the airfoil section is even more difficult.

The airfoil is modeled with a large series of Bezier curves and is a complex surface. However, the topology of a turbine blade does not drastically change. This is why the turbine blade is a perfect candidate for a design tool that incorporates the regenerative approach.



Figure 3: Turbine Blade, Courtesy of Pratt & Whitney

3.3.2 Feature Library Approach

The feature library approach involves taking previously drawn part features and combining them to create an entirely new product. These are customized, complex user defined features. For this approach, the user specifies what features they want, the amount of each feature, the placement of the features, and the geometric dimensions for each feature. A program is then run that combines all the features according to the user specifications.

There are several positive aspects to the feature library approach. It is easy for the user to make a topology change. Adding or deleting a feature from the model quickly makes a topology modification. Furthermore, to make a small change to the geometry, only one feature needs to be modified. Thus, the entire part does not need to be redrawn and it is quick to make small changes. The last advantage to this approach is that engineers can expand the tool. Generating new features, which can be done by anyone who can interactively use the CAD software, expands the tool to include new designs. . Thus, the tool can improve without the involvement of a programmer. This is a useful advantage because design tools become obsolete if they cannot be modified to reflect new technologies and designs.

There are problems with the feature library method. This approach creates more work for the user. This causes the design tool to have a higher probability of failure. For example, the user controls where the features exist in space and it is possible for the user to position features such that they do not form a solid geometry. Another drawback is that it is hard to handle complex geometries with the feature library approach. For example, it is not easy to add a fillet to an edge that is created by the interaction of two separate features. It cannot be guaranteed that the edge exists and would be difficult to find that particular edge from a programming standpoint. Any geometry that is complex needs to be completely contained within one feature. Geometry that involves the interaction of two separate features needs to be added manually at the end of the design process. Clearly, not every part should utilize the feature library approach.

This research utilizes the feature library approach to develop a design tool for BOAS. The feature library approach is the better choice for BOAS, because the design tool needs to be able to create multiple BOAS designs. Figure 4 shows various examples of BOAS.

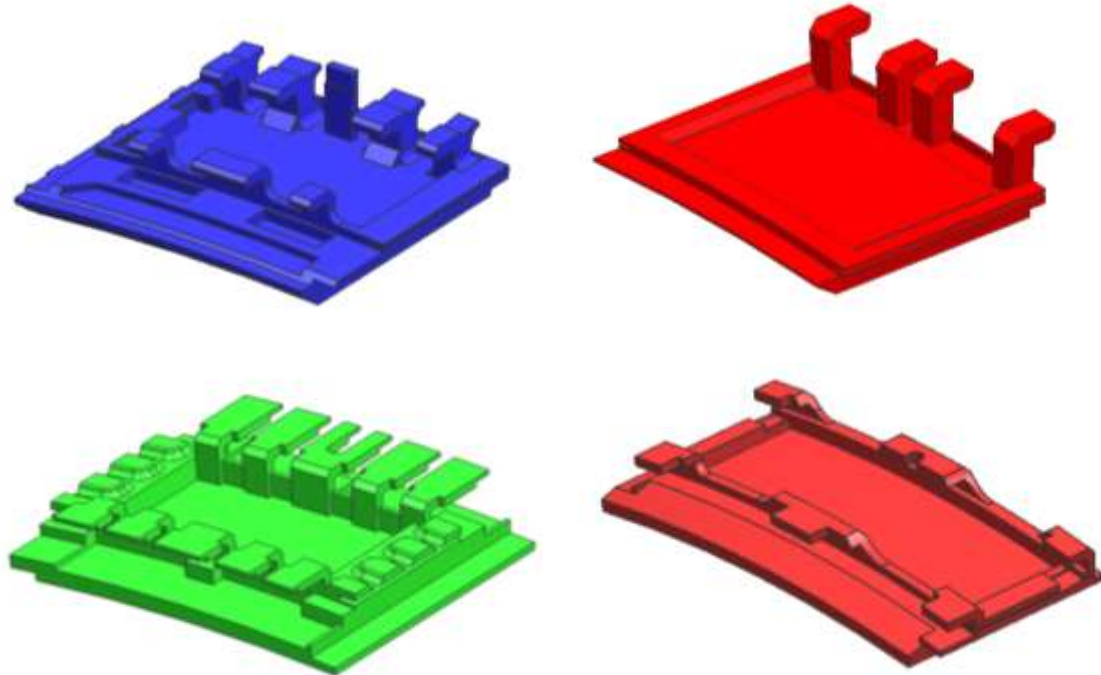


Figure 4: Representative Examples of BOAS

These four BOAS represent four drastically different topologies. However, each BOAS is not overly complicated, especially when compared to the turbine blade. The BOAS are mostly extrusions and revolutions, which are less complicated than Bezier splines. Since the geometry is not excessively complicated and the topologies are quite different, the feature library approach is ideal for a BOAS design tool.

It is important to capture the various BOAS topologies. The BOAS are divided into five basic features; base, hook, shiplap, rail, and core. Other miscellaneous features include cover plates and anti-rotation features. All of these features come together to form a complete BOAS. Figure 5 depicts how different features come together to create a BOAS.

3.4 Features

The first step to develop a feature library based design tool is to create the feature library. This can be done interactively using CAD software. For this project, Unigraphics[®] was used. The key to having a quality set of features in the library is having the features created parametrically. This allows the features to be updated with geometric changes. If a feature does not have a valid parametric scheme, nonsensical or nonmeshable geometry may be created. One solid body could be split into multiple bodies or tiny faces and slivers could be created. These are all results that would cause meshing problems later on or cause the geometry to fail in Unigraphics[®]. The input parameters are also important. There can be many ways to parameterize a feature but they are not all equal. Some schemes may make more sense for the users and may more accurately reflect how a design typically changes. The flexibility versus automation problem is prevalent at this point. The designer could generate a scheme where only a few parameters are specified and most are calculated. This

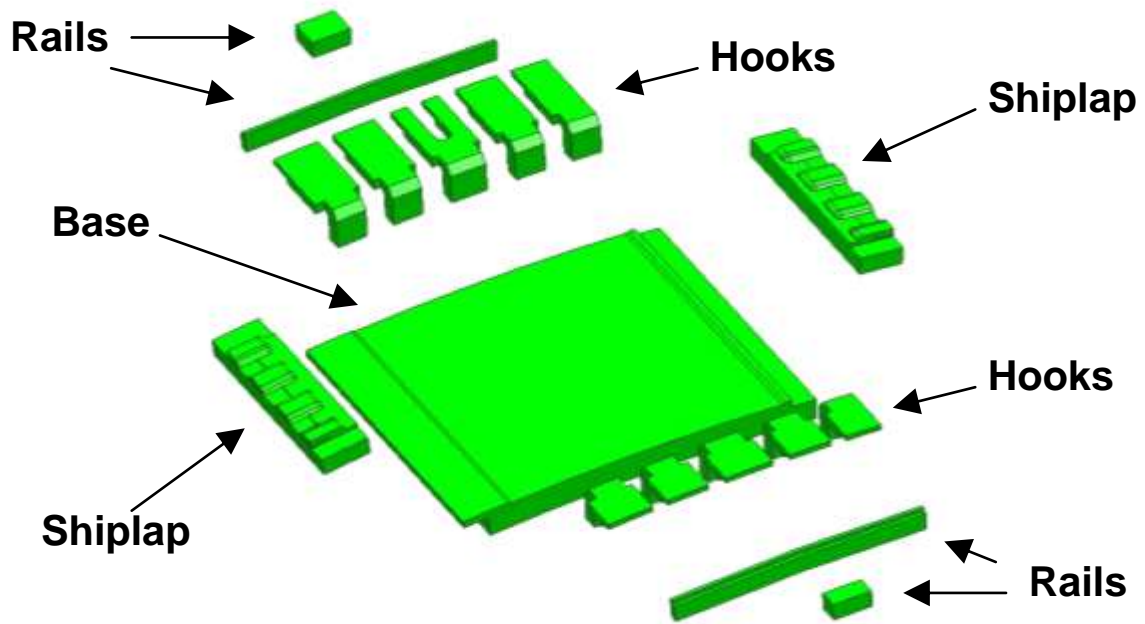


Figure 5: Features form a BOAS

makes it easier for the user since there is more automation. Another option is to have a scheme where many parameters are specified. This gives the user more control and is a more flexible design. Having good parametric features that fit the user's need is the foundation of a feature library system.

Another important aspect of the features is that they have named attributes. The surfaces, volumes, and edges of the features need to have specific names. This is important for the downstream analysis process. The names of these attributes can be understood by analysis software. This makes the application of boundary conditions and mesh refinement much easier.

To start the feature library for this project, four previous BOAS designs were chosen to be included in the feature library. All of the features necessary to completely create these BOAS were developed. These four BOAS reflect different topologies and the latest designs. Thus, the developed features represent an array of design alternatives and it is likely that they will be reused for future designs. Furthermore, not all of the details are included in the features. This geometry is intended to facilitate engineering analysis. Some details are not important for the analysis and including them slows down the analysis. Thus, the important design details are captured but those that are not important are left out. For example, details, such as fillets, are included in expected areas of failure but are not included elsewhere. Expert opinion is used to make this determination. If there are areas of failure that are not expected, detail is added to those areas and the model is rerun.

3.5 Unigraphics® Design Tool Program

Programming is the driver of the entire process. This program is written using the Unigraphics® Application Program Interface (API), which is C++ with functions specific to Unigraphics®. The program is written and compiled with Microsoft Visual Studio®. This API allows all the functionality of the Unigraphics® Graphical User Interface (GUI) to be

included in a program. The design tool program has two jobs. It imports the user features into a seed file, which is a standard Unigraphics® part file, and modifies the features within the seed file. This is implemented through the use of text files. The user specifies what features they want to import and the associated feature parameters. The program is run and the geometry is imported. The text files are then modified by the user and the program is rerun to alter the geometry. The process is repeated until the desired geometric model is produced. The final design must be a robust parametric model so it is simple to make small changes. This process is shown in figure 6.

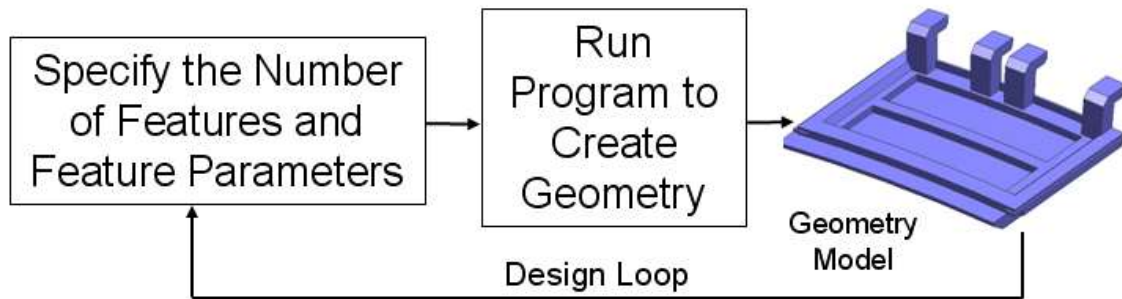


Figure 6: Developed Design Process

3.6 Benefits of the Developed Process

The design tool creates a new design process that allows the user to produce parametric and analysis capable geometry faster than under the traditional approach.

3.6.1 Traditional Process

The traditional process is the typical design approach that is in use before a design tool is developed. This conventional process takes too long. It involved someone manually creating a new model from scratch. This new design is not created to be parametric. Thus, it is difficult to make modifications to the geometry. Simple morphing is difficult and topology changes are even harder. Furthermore, there is no assurance that the geometry is meshable. Thus, even though the design is of the desired configuration, the engineers may not be able to conduct an analysis with the computer model. This process makes it impossible for engineers to cycle through many design iterations before deciding on a final design.

3.6.2 Developed Process

The new process is much quicker than the traditional one. The user can start with a previous design or use features from an existing design to start their new BOAS. From there, the user just iterates through the design process shown in figure 6 until the desired model is produced. They only have to create the new features that they need, rather than a whole BOAS. Furthermore, the entire feature library is parametric so it is easy to make morphing changes to the geometry. Also, since there is a feature library, it is simple to make topology changes by adding and deleting features. Another advantage is that all of the features are certified to be meshable. The features are tested to ensure meshability. Therefore, the user only has to make sure their specific combination of features is meshable. Also, any new features they develop must be tested for meshability. Once that is done, any other user can utilize those features. This new process allows the engineers to create many different designs before a final design is selected.

4 Analysis Tool with ANSYS® Academic Research

Employing an analysis tool creates a new analysis process that is faster and easier to use than the traditional analysis process.

4.1 Goals of an Analysis Tool

The goals of an analysis tool are similar to those of the design tool. First, a geometric model must be the only required item before the analysis tool can be run. The user should only have to run the analysis tool to get the results and there should be no interaction with the analysis software. The analysis process must be completely automated. Therefore, it is simple to use. Second, automating the analysis process has to save time. Manually applying boundary conditions can be tedious if there are a large number of faces with different boundary conditions. Automating the analysis eliminates this time consuming process. Finally, automating the analysis process must improve the quality of the analysis. An engineer could easily make a mistake preparing the finite element model for analysis. An automated tool never makes a mistake once it is bug free.

4.2.1 Development Approach

To create an automated analysis tool, an analysis program was written for ANSYS® with the ANSYS® Parametric Design Language (APDL). All the functionality of the ANSYS® GUI is available in the programming language. Thus, the developed program developed mimics what an engineer would perform manually. There are several basic steps in the analysis process and these are shown in Table 1.

Analysis Process Steps
1 Import Unigraphics® Geometry in ANSYS®
2 Apply Material Properties
3 Mesh the Geometry
4 Apply Boundary Conditions
5 Run Analysis
6 Display and Store Results

Table 1: Analysis Process Steps

All of the steps shown in table 1 can be done by a computer program written using the APDL. Several analysis tools that follow this basic procedure have been previously created. For example, an analysis tool that follows the basic steps in table 1 was developed for turbine blades. Thus, it is simpler to modify the existing analysis tool for BOAS geometry rather than to recreate previous work. The existing program was changed to import BOAS geometry and to recognize BOAS attributes. Furthermore, a different set of boundary conditions need to be applied and new results had to be calculated. In essence, turbine blade analysis tool provided the basic framework and it was adapted to be BOAS specific.

4.3 Developed Analysis Process

The developed process aims to automate the entire analysis process. Instead of having the user interact with ANSYS®, a computer program does all the work. Furthermore,

all analyses are integrated seamlessly. Previously, thermal and structural analyses were conducted separately on a BOAS. With the automated process, there is no distinction between the two analyses. The user runs the programs and all the results, whether they are thermal or structural, are reported once the program has finished. The new process is shown in figure 7.

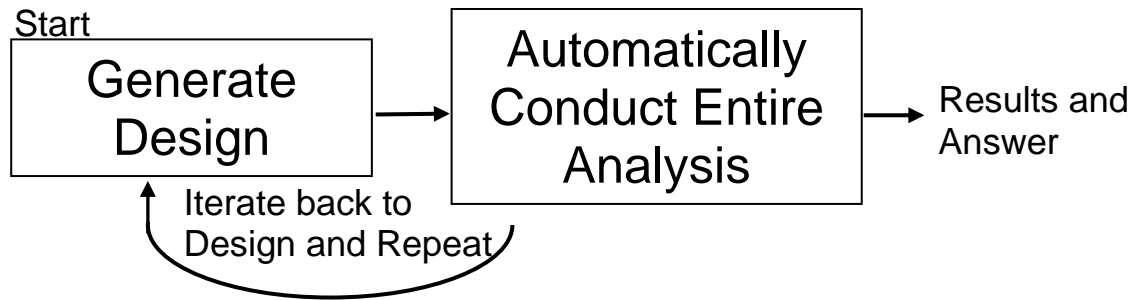


Figure 7: New Analysis Process

There are several benefits to this new process, which lead to the engineers spending more time engineering and less time completing repetitive tasks. The user no longer needs to prepare a finite element model for analysis or be adept with the FEA software. They just need to run a program and interpret the results, which they now have more time to do. With more time to evaluate the results, the user will be able to research phenomena that they did not have the time to do before. This can lead to improved quality since the engineer can now delve into both primary and secondary results. Furthermore, more design-analysis iterations can be run and more designs can be attempted. This gives the engineers freedom to consider many different design configurations rather than just one or two. This leads to improved quality and it allows for optimizations and sensitivity analyses. These are powerful statistical tools for developing quality parts and understanding the important factors in the design. Essentially, the engineers have more ability because the analysis tool eliminates time consuming analysis tasks.

5 Design-Analysis Link with i-Sight FD

Integrating the design and analysis tools provides the engineers with new capabilities that allow them to generate higher quality products.

5.1 Goals of a Linked Design-Analysis Tool

Having separate design and analysis tools is powerful for an engineer. However, having the two tools linked together as an automation loop is a much more potent tool. With the two tools linked together, only the design parameters need to be specified for each design. Thus, engineers can create multiple design parameter sets. The linked loop then generates the geometry, analyzes the model, and stores the results for each parameter set. This allows for multiple designs to be analyzed without the constant involvement of an engineer. Achieving this goal unlocks the ability to conduct sensitivity analyses and optimizations.

5.2 Development Approach

The first part of the automation loop is geometry creation. The design tool was used to create a fully parametric model of the product. It is this parametric model that is used in the automation loop. The parametric model is flexed to reflect the current set of design parameters. The second step is the analysis. For this, the actual analysis tool is implemented. The geometric model is imported into ANSYS® and analyzed. To communicate information between the parametric model and ANSYS®, feature attributes are converted into ANSYS® component sets by using the table correspondence file. Finally, the results for all design iterations are stored together.

i-Sight FD®, developed by Eneignous Software, is optimization software that can create this automation loop. First, the user specifies a list of design points in a text file. For each design point, the values of several design parameters are given. i-Sight FD® reads this text file and imports the design parameter values into the parametric model. This updates the geometry and a new design is generated. The analysis tool then imports the geometry and produces FEA results. Finally, the results are stored in a text file. i-Sight FD® then moves to the next design point and those results are stored with the previous results. This process is repeated until all design points are analyzed and their associated results are stored. The automation loop flow chart is shown in figure 8.

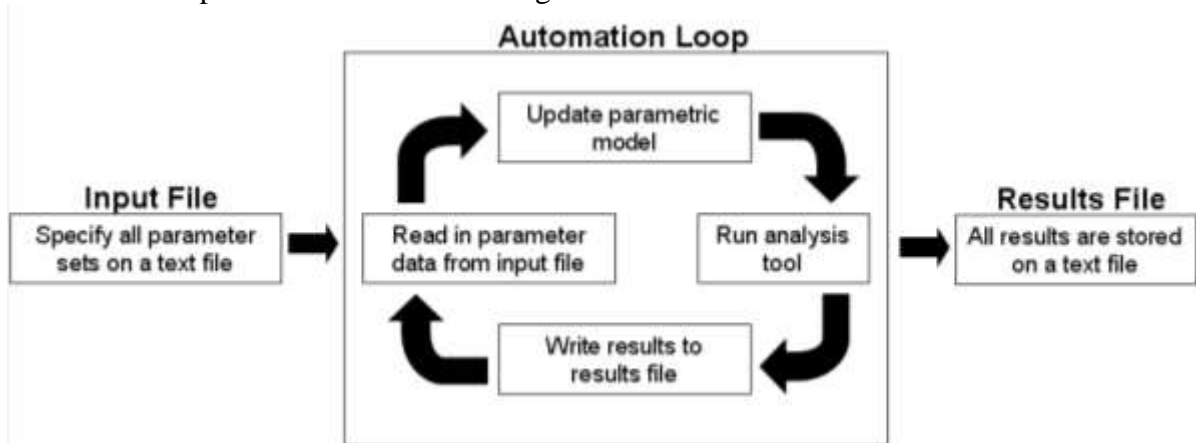


Figure 8: Automation Loop

There are two tools provided by i-Sight FD® that make this automation loop possible. The first tool is the file parsing component. This allows i-Sight FD® to write to and read from text files. This is important for updating the parametric model. All the geometric parameters that define the geometry can be listed in an expression file, which is a basic text file. If the values on the expression file are changed and then imported into Unigraphics®, the parametric model is updated. This is how the geometry is updated for each design point. i-Sight FD® utilizes the file parsing component to modify the expression file. The i-Sight FD® second tool, the command line tool, is then used to launch Unigraphics® in batch mode. This launches Unigraphics® without its GUI. This is when the expression file is imported into Unigraphics® to update the geometry. Another program was written using Unigraphics® API to accomplish this task. The command line tool runs Unigraphics® and this external Unigraphics® program updates the geometry.

The two tools are also utilized with ANSYS®. The command line tool launches ANSYS® and runs the analysis tool. Once the analysis is complete, the results are written to a text file. This file is read with the i-Sight FD® file parsing component and the results are all

stored together in a single text file. These two tools provided by i-Sight FD[®] allow automation loops to be created.

5.3 Benefits of an Automation Loop

With this process in place, the engineer can now quickly analyze many different design configurations. The engineer only needs to create the design point text file and start the automation loop. Once all the results have been calculated, which could take days on a single computer, an engineer can conduct a sensitivity analysis or an optimization. They can also find areas of further exploration from the results. Furthermore, if the process is run on a computer that is not the engineer's workstation, then the engineer is free to complete other tasks. This improves the productivity of the engineer. Finally, this automation loop improves quality and is cost effective because more design-analysis cycles can be completed with less involvement from the engineer.

5.4 Example of Process

A design tool, analysis tool, and automation loop were developed for BOAS as part of this project. With these tools, a sensitivity analysis was conducted to determine what parameters are most important in terms of alloy life, which is the time it takes for oxidation on the inner diameter surface to cause a BOAS to fail.

Two of the parameters, Core Radial Placement and Core Radial Thickness, control all the cores. Thus, each core has the same Core Radial Thickness value. For Core Radial Placement, there are two sets of cores, the inner and outer cores. The inner cores have a fixed offset distance from the outer cores. Therefore, all the cores move together as Core Radial Placement changes and the fixed offset is maintained. ID Grind controls the inner diameter surface relative to the reference plane and Panel Placement controls the panel position relative to the reference plane.

Next, upper and lower bounds are placed on each input parameter and the resulting design space is filled with a Latin Hypercube (Santner et al., 2003). The automation loop is utilized to analyze each design point. Figure 9 shows the robust parametric model in Unigraphics[®] and its associated contour plot that is generated in ANSYS[®].

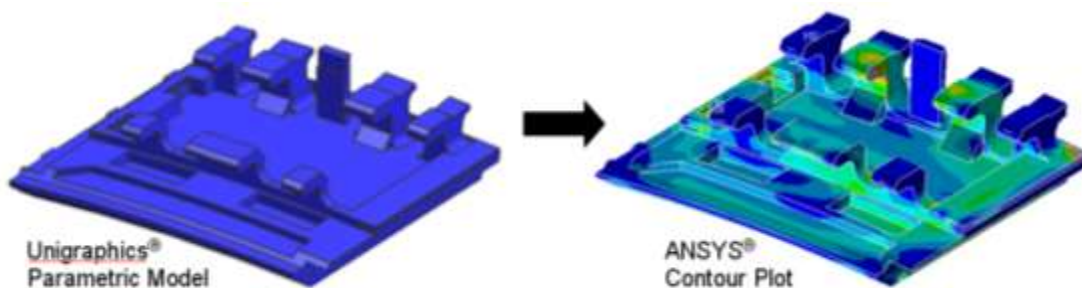


Figure 9: Robust Parametric Model and Corresponding Results

The parametric model and analysis tool are contained within the automation loop. This loop is shown in figure 10 and depicts the process that was run in i-Sight FD[®].

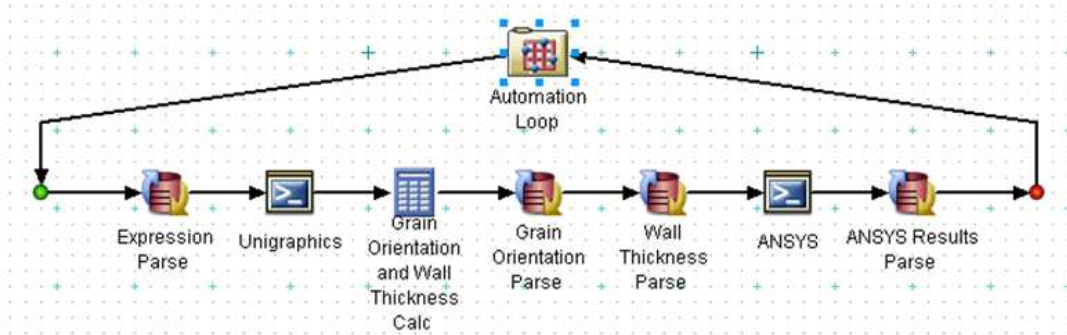


Figure 10: i-Sight FD[®] Automation Loop

This loop updates the parametric geometry model, runs the analysis, and stores the results for each design point on a single text file. Once all the data was gathered, a Pareto plot was generated by using Gaussian Emulation Machine for Sensitivity Analysis (GEM-SA) software developed by Marc Kennedy (Kennedy, O’Hagan, 2006). This plot is shown in figure 11.

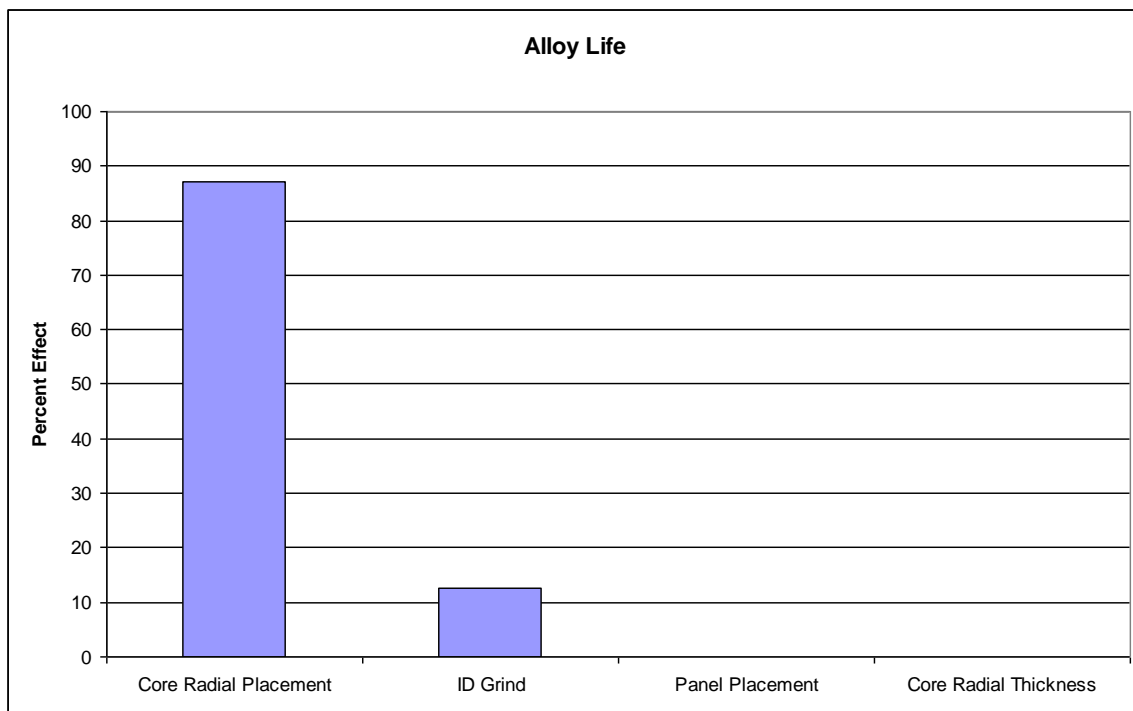


Figure 11: Pareto Plot of Alloy Life

This figure depicts that Core Radial Placement and ID Grind are the two most important factors in terms of alloy life. Therefore, engineers should concentrate on Core Radial Placement and ID Grind when design a BOAS in the future. It should be noted that these results could not be obtained without the MDO approach and the software developed to implement it. It would have taken too long to conduct the automation loop manually and too expensive to run the physical experiments.

6 Conclusions

This paper demonstrates the ability of commercial software to successfully implement Multi-Disciplinary Optimization (MDO). A Blade Outer Air Seal (BOAS) design tool was developed by using the Unigraphics[®] Application Program Interface (API), which is C++ with functions specific to Unigraphics[®]. Next, an analysis tool for ANSYS[®] was created by utilizing the ANSYS[®] Parametric Design Language (APDL). This tool included both thermal and structural analysis capabilities. Finally, i-Sight FD[®] was used to link the two tools into an automation loop. This coupled design and analysis loop provides engineers with the ability to conduct MDO. To prove that MDO can be successfully implemented, a Variance Based Sensitivity Analysis (VBSA) was conducted (Chan et al.). To be able to run a VBSA, the parametric model, API programming, and i-Sight FD[®] automation loop must be robust. Thus, running a VBSA proves that commercial software is indeed robust enough and can be integrated together to implement MDO. Therefore, the tools developed for BOAS can be used for MDO.

The results obtained from the VBSA gives engineers a better understanding of the problem. In terms of alloy life, the engineers now know which factors are the most important. Thus, when the next BOAS is designed, the engineers can focus on other important factors. Furthermore, this is the type of information that can only come from implementing MDO. Without the computer tools that were developed to employ MDO, the VBSA could not have been conducted. It would be too expensive to run enough physical experiments and too time consuming to design and analyze each design manually. Clearly, MDO allows engineers to gather information that would be feasibly unattainable without it.

Acknowledgements

A special thanks to Pratt & Whitney for their support and expertise. They provided programs that were used and modified for this work. Furthermore, Pratt & Whitney employees answered technical questions and offered their insights on the proper method to develop the design, analysis, and automation tools. Finally, thanks to ANSYS Inc. for providing ANSYS[®] Academic Research v11.0.

References

Anderl R., and Mendgen R., "Parametric Design and its Impact on Solid Modeling Applications," *Proceedings of the 3rd Symposium on Solid Modeling and Applications*, Salt Lake City, Utah, December 1995, pp. 1-12.

ANSYS[®] Academic Research, v 11.0

Ault H., "Using Geometric Constraints to Capture Design Intent," *Journal for Geometry and Graphics*, Vol. 1, No. 1, 1999, pp. 39-45.

- Bartholomew, P., "The Role of MDO Within Aerospace Design and Progress Towards an MDO Capability," AIAA Paper 98-4705, 7th AIAA/USAF/NASA/ISSMO Symposium on Multi-disciplinary Analysis and Optimization, St. Louis, MO, September 1998.
- Bidarra, R., and Bronsvort, W. F., "Semantic Feature Modelling," *Computer-Aided Design*, Vol. 32, 2000, pp. 201-225.
- Bralla, J. G., *Design for Excellence*, USA: McGraw-Hill, 1996.
- Chamis, C. C., "Coupled Multidisciplinary Optimization of Engine Structural Performance," *Journal of Aircraft*, Vol. 36, No.1, 1999, pp. 190-199.
- Chan, K., Tarantola, S., and Saltelli, A., Variance-Based Methods. In Saltelli, A., Chan, K., and Scott, E., Sensitivity Analysis, UK: John Wiley & Sons Ltd, 2000.
- Cunningham, J., and Dixon, J. R., "Design with Features: The Origin of Features," *ASME Computers in Engineering Conference*, San Francisco, July 31- August 4, ASME Press, 1988, pp. 237-243.
- Cutkosdy, M., Tenenbaum, J. M., and Muller, D., "Features in Process Based Design," *ASME Computers in Engineering Conference*, San Francisco, July 31- August 4, ASME Press, 1988, pp. 557-562.
- Giesing, J. P., "A Summary of Industry MDO Applications and Needs," 7th AIAA/USAF/NASA/ISSMO Symposium on Multi-disciplinary Analysis and Optimization, St. Louis, MO, September 1998.
- Hardee, E., Chang, K., Tu, J., Choi, K., Grindeanu, I., and Yu, X., "A CAD-based Design Parameterization for Shape Optimization of Elastic Solids," *Advances in Engineering Software*, 1999, Vol. 30, pp. 185-199.
- Hoffmann C., and Kim K., "Towards Valid Parametric CAD Models," *Computer-Aided Design*, Vol. 33, 2001, pp. 81-90.
- Hogge, D. G., Integrating Commercial CAx Software to Perform Multidisciplinary Design Optimization, M.S. Thesis, Brigham Young University, 2002.
- Jones, C. L., *Parametric Model Behavior Across Multiple CAD Systems*, M.S. Thesis, Brigham Young University, 2001.
- Kennedy, M., O'Hagan T., <http://www.tonyohagan.co.uk/academic/GEM>, 2006.
- Lewis, K., and Mistree, F., "The Other Side of Multidisciplinary Design Optimization: Accommodating a Multiobjective, Uncertain, and Non-Deterministic World," *Engineering Optimization*, Vol. 31, 1998, pp. 161-189.

- Miner, R. H., A Method for the Representation and Manipulation of Geometric Features in a Solid Model, M.S. Thesis, MIT 1985.
- Prasad, B., "Product Planning Optimization Using Quality Function Deployment", in *AI in Optimal Design & Manufacturing*,. Z. Dong, (ed.) and series ed. Mo. Jamshidi. Englewood, NJ: Prentice Hall, 1993, pp. 117-152.
- Prasad, B., Competitiveness Analysis of Early Product Introduction and Technology Insertion, Proceedings of the 1994 Int'l Mechanical Engineering Congress and Exposition, Nov. 1994, Chicago, IL. PED-Vol. 68-1, *Manufacturing Science and Engineering*, Col. 1, ASME, 1994, pp. 121-134.
- Prasad, B., "A Structured Approach to Product and Process Optimization for Manufacturing and Service Industries," *International Journal of Quality and Reliability Management*, Vol. 12, No. 9, 1995.
- Pratt, M. J., and Wilson, P. R., "Conceptual Design of a Feature-Oriented Solid Modeler," *Draft Document 3B*, General Electric Corporate R&D.
- Rohl, P., He, B., and Finnigan, P., "A Collaborative Optimization Environment for Turbine Development," AIAA Paper 98-4734, 7th *AIAA/USAF/NASA/ISSMO Symposium on Multi-disciplinary Analysis and Optimization*, St. Louis, MO, September 1998.
- Santner, T., Williams, B., Notz, W., *The Design and Analysis of Computer Experiments*, USA: Springer, 2003.
- Sevenler, K., Sherman, M. K. and Vidal, R., Multidisciplinary Teamwork In Product Design: Some Requirement for Computer Systems, *ICED*, 1993, pp. 343.
- Sobieszanski-Sobieski, J., and Haftka, R., "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments," *Structural Optimization*, Vol. 14, 1997, pp. 1-23.
- Srinivsan, H., Jensen, C.G., Kopper, F.C., Staubach, J. B., and Pack, D. R., "Assembly Parametrics for Multidisciplinary Design Optimization," 8th *International Society for Productivity Enhancement Conference on Concurrent Engineering*, Anaheim, CA, July 29 – August 1, 2001, pp. 158–165.
- Subrahmanyam, S., DeVries, W., and Pratt, M. J., "Feature Attributes and Their Role in Product Modeling," *Proceedings of the 3rd Symposium on Solid Modeling and Applications*, Salt Lake City, Utah, December 1995, pp. 115-124.
- Tappeta, R. V., Nagendra, S., and Renaud, J. E., "A Multidisciplinary Design Optimization Approach for High Temperature Aircraft Engine Components," *Structural Optimization*, Vol. 18, 1999, pp. 134-145.

Townsend, J., Samareh, J., Weston, and R., Zorumski, W., "Integration of a CAD System into an MDO Framework," NASA TM-207672, May 1998.

Tucker, S. S., *Parametric Application Program Interface Function Mapping of Computer-Aided Design Systems*, M.S. Thesis, Brigham Young University, 2000.

Turner, G., and Anderson, D. C., "An Object Oriented Approach to Interactive, Feature Based Design for Quick Turnaround Manufacturing," *ASME Computers in Engineering Conference*, San Francisco, July 31- August 4, ASME Press, 1988.

United Technologies- Pratt & Whitney, "PW4000 94-Inch Fan Engine," <http://www.pw.utc.com/StaticFiles/Pratt%20&%20Whitney%20New/Media%20Center/Assets/1%20Static%20Files/Images/pw400094_cutaway_high.jpg>

Zhou, X., Yanjie, Q., Guangru, H., Huifeng, W., Xueyu, R., "A feasible approach to the integration of CAD and CAPP," *Computer-Aided Design*, Vol. 39, 2007, pp. 324-338.