# Using Knowledge-based Solid Modeling Techniques and Airfoil Design Data: A Case Study in Developing an Airfoil Seed Part Generator

William D. O'Brien, Michael Chandonais, R. Scott Henderson, Mark Prezbindowski, Nathan W. Hartman, Joseph A. Rasche

Purdue University (obrienw@purdue.edu) / Purdue University / Purdue University / Purdue University / Purdue University (nhartman@purdue.edu) / Rolls-Royce Corporation (joe.a.rasche@rolls-royce.com)

## Abstract

This paper details a research project which encompassed the development of corporate best practices surrounding the use of company specific compressor and turbine design data in the generation of solid model geometry. Current solid modeling processes at Rolls-Royce Corporation for compressor and turbine engine airfoil geometry require a great deal of time and are prone to human error in the replacement of CAD geometry when design changes are made. A team of Purdue University students and faculty collaborated with Rolls-Royce Corporation to improve this process using knowledge-based tools within the Unigraphics (UG) NX 2 modeling tool set. A graphical user interface to import and manipulate aero data was developed and tested to ensure that it met design standards and maintained airfoil geometric integrity within the Rolls-Royce environment. Times for airfoil model replacement using the new application were recorded and compared to the previous method. Results demonstrated that the knowledge-based modeling applications had an average of 88.3% time savings as well as allowed for automation of the solid model creation process of the airfoils. Suggestions for integration of such techniques into engineering and technology education are also presented, along with implications for education and industry.

## Introduction

This project represents a project specification undertaken by four students in a senior project course (CGT 411) in the Computer Graphics Technology (CGT) department at Purdue University. This course provides students the opportunity to work in a team-based environment to complete a project for an outside client. The client can represent any combination of the four areas of the CGT department: virtual product integration, construction graphics management, interactive media and imaging, and animation and gaming. Three of the four students on this project were specializing in the virtual product integration area, while the fourth student specialized in interactive media and imaging. They brought a unique blend of technical skills and competencies that included 3D geometric modeling (surface and solid), scripting and programming, technical writing, and prior industrial experience. One of the goals of the CGT 411 course is to have student teams use all of their knowledge acquired in their previous scholastic experience and develop a solution to a client-based problem. In doing so, the students establish a fictional company for doing the work and administer the work to be done, including financial reports, business plans, staffing, and legal issues. A substantial portion of the course grade is based on how well they accomplish these things, in addition for developing a research strategy to determine the effectiveness of their problem solution. The remainder of this paper details the experiences of four CGT students in the Spring 2006 semester working on a project with Rolls-Royce Corporation – Indianapolis.

The design of turbine engines at Rolls-Royce Corporation involves the modeling of airfoil blades. This is a very iterative process centered on the analysis and revision of airfoil CAD models. After revisions are made, geometrical data is re-engineered and recreated within Unigraphics NX2. This process ranges from hours to days because the current methods of creating the airfoil models in the CAD system are not parametric, i.e. the geometry is not associated with the engineering definition of the airfoil after the model is created. An alternative method of creating the geometry has been developed using Knowledge Fusion (KF), a scripting language within NX2. This new method kept the airfoil geometry associated to the engineering airfoil definition which allowed Rolls-Royce modelers to replace the airfoil within the existing engine rather than deleting geometry and rebuilding it. A graphical user interface (GUI) was developed using the User Interface Styler within NX2 to easily interact with the Knowledge Fusion plug-in. By incorporating existing knowledge of design rationale and intent into an easy-to-use plug-in, geometric models of airfoils can be created with the click of a button. The resulting models are feature-based and parametric, allowing modifications to be made automatically through the application or later manually within NX2, resulting in 88.3% time savings. A glossary is included at the end of this paper to aid the reader in understanding the technical terminology involved with this project.

## Overview of Airfoil Design

Airfoil blades are a crucial component within a turbine engine, and their design covers many engineering disciplines such as thermodynamics and statics. For both analysis and manufacturing these airfoils are modeled in a CAD system. However, the complex shapes of airfoils make this difficult. They are typically modeled using b-splines or NURBS and the development of methods to do this has been ongoing for decades [1, 2]. At Rolls-Royce Corporation the definition of an airfoil is determined by custom computer programs that take engineering parameters and compute points at various sections of a potential airfoil. The points are saved in a text file (known as an airfoil definition) for modeling. After the airfoil itself is modeled, the geometry that is used to attach the airfoil to the engine (the stalk and platform) is added to the airfoil. Figure 1 shows an example of an airfoil.

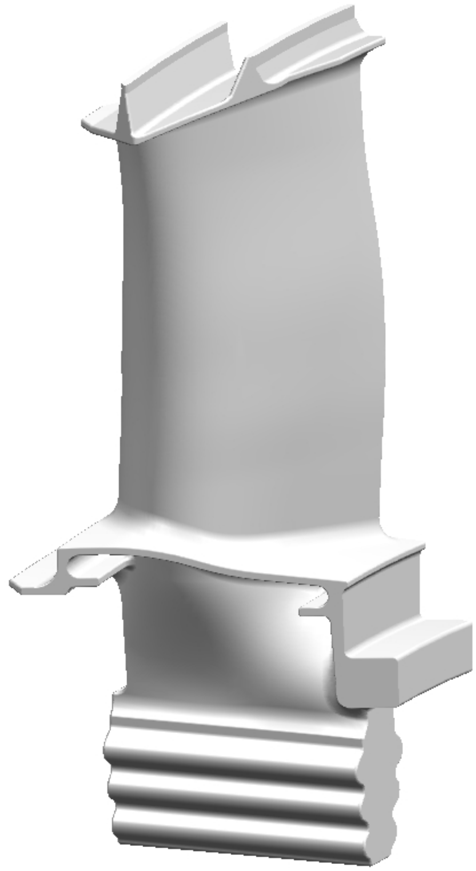

Figure 1: Airfoil with stalk (Rolls-Royce Corporation)

If the airfoil is a turbine blade, as opposed to a compressor blade, additional geometry is created to hollow out the airfoil and add air passages that are used to cool the airfoil while the engine is running. Due to the geometric complexity of these features, compressor blades were used as a starting point for the evaluation of this modeling technique. After an airfoil model is completed it is then analyzed by computer programs to determine if it meets its design requirements. Various analyses, such as FEA and CFD, are run to determine the properties of the airfoil such as strength, vibration, and thermal characteristics. Based on the results of these analyses, engineers determine a new airfoil design and the process repeats - often dozens of times for each airfoil. For compressor blades, each airfoil iteration requires 1 – 2 hours using the existing methods described previously. For turbine blades, each iteration requires 8 – 16 hours using existing methods. These values illustrate the need to explore alternative methods for iterating the design of airfoils and the use of CAD geometry in the process.

A turbine engine can contain upwards of twenty different airfoils, so any improvement in the process of one design iteration will have a beneficial effect on the total design process. Rolls-Royce Corporation uses the Unigraphics NX2 CAD package as its primary modeler and has developed several methods of creating airfoil models from the airfoil definitions. All but one of the current methods for airfoil modeling in NX2 do not create parametric airfoils [3]. The exception is a program used in the German division of Rolls-Royce; however, that program does not conform to U.S. design standards. This means that after the initial geometry is made, no links to an airfoil definition exist. For every design iteration the complete airfoil model must be torn down, the airfoil recreated, and the model rebuilt. In some instances, such as the much more complex turbine airfoil, the entire model may have to be recreated.

Knowledge-Based Engineering (KBE) "has gradually gained prominence as a major tool to speed up product development…" [4]. Essentially, KBE is the capturing of knowledge from designers and engineers and automating it within a program [4, 5, 6, 7]. This knowledge is then used to assist designers while they create products within the CAD system [8]. KBE systems are capable of automatically creating objects [9, 10], assisting designers while they create objects [11], and comparing the cost versus efficiency of created objects [12, 13]. There are many examples of companies that successfully used Knowledge-Based engineering systems for the development of their products. For example, Park et al. [13] developed a method that produced multiple variations of blade designs for a VTOL aircraft, performed finite element analyses, and then chose the optimal blade based on the results. The ability of KBE systems to reduce design and analysis time is an optimal reason why these systems are being implemented in companies around the world [14]. In a case closely related to the Rolls-Royce scenario, an aerospace company implemented a system which allowed them to reduce design iteration from 80 hours to four hours [15].

Unigraphics (UGS) NX2 (and other versions) contains several methods of modeling automation which are already integrated with NX and included with the software. These include macros, GRIP, Knowledge Fusion (KF) and NX/Open. Macros and GRIP have the lowest level of functionality and no associativity with the geometry they create [16]. NX/Open has the greatest capability but also requires the most specialized knowledge of any of the tools and has less associativity than KF [16]. Knowledge Fusion and NX/Open can be combined and also interact with the Unigraphics' TeamCenter PDM software [16, 7].

KF is an object-oriented, hierarchical, declarative and rule driven scripting language for KBE [7]. Geometric features are available as classes which allow for easy creation of dynamic geometry. A KF developer is able to create custom "applications" for NX that provide the ability to modify characteristics of the product without having to perform a complete redesign. Widespread usage of KF appears limited. As noted by Carleton [11]:

> Although the knowledge rules were relatively easy to apply to geometry during a design, a high degree of training was involved in developing such rules as they require following a strict procedure. (p. 38)

Another turbine engine manufacturer worked with UGS consultants to develop an intelligent master model of an entire aircraft engine heavily based on KF. They developed this in order to reduce iteration time and increase collaboration between component specialists [17]. By using GUIs (Graphical User Interfaces), designers were then able to change components of the engine so that new versions of that engine were created. However because of the advantages of KF as discussed and a proof-of-concept study conducted by Rolls-Royce the decision was made to use KF to improve the airfoil design process. By using this KBE technology, the time required to remodel an airfoil for each design iteration was reduced. Previous design iterations required the tear-down and rebuilding of the airfoil model for each airfoil definition while the airfoil based on KF remained associative to the airfoil definition. This meant that a new airfoil definition could be specified and the model would update accordingly. Reducing the time of a single design iteration created a significant total time savings for the entire design process.

**Application Overview**

In order to meet the needs of Rolls-Royce, an NX2 plug-in was developed using KF to read and interpret ASCII files containing airfoil definitions and generate the geometry described by the 3D information contained in the files. KF technology within NX is an object-oriented scripting utility. In KF, each object class was written as an individual text file (.dfa file extension), using the included NX object classes as a starting point. The NX plug-in created during this project was designated the Airfoil Seed Part Generator and was divided into compressor and turbine components. A simplified hierarchy of these classes used in the compressor component of the Airfoil Seed Part Generator is as follows:

- RR_Compressor
  - RR_Parser_XDC
  - RR_Parser_RRAF
  - RR_DC_Flowpath / RR_Flowpath
  - RR_SectionSplines
  - RR_CapLines
  - RR_Caps
- RR_RadialSplines
- RR_FaceSurfaces

**RR_Compressor** served as a parent, or container, class for the compressor component. It contained a number of child classes responsible for the creation of individual elements of the airfoil geometry, as well as the code that enabled interaction with the GUI. The child classes **RR_Parser_XDC** or **RR_Parser_RRAF** were used to open and parse an airfoil definition file. Airfoil definitions were provided to modelers in an ASCII text file using the XDC or RRAF format developed by Rolls-Royce Corporation. **RR_Flowpath** drew a partial flow path if one was contained in the XDC file. The **RR_DC_Flowpath** class optionally parsed an external flow path definition file and generated the resultant paths (see Figures 2 and 3).

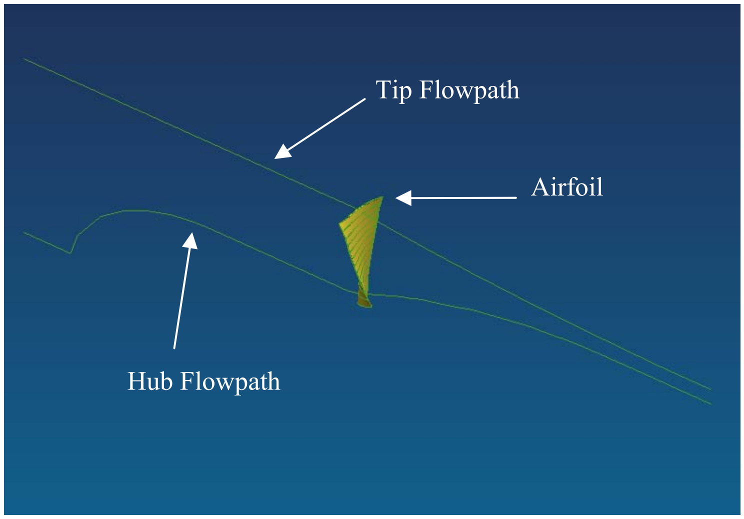


Figure 2: Airfoil model with flow paths

**RR_SectionSplines** drew each of the stacked sections of the airfoil while **RR_RadialSplines** drew the four radial splines at the tangency points. The geometry of the two end caps (leading and trailing edges) was created by **RR_Caps** and **RR_CapLines.** Finally, **RR_FaceSurfaces** created four surfaces: the leading and trailing edges and the pressure and suction sides of the airfoil. These features are illustrated in Figure 3.

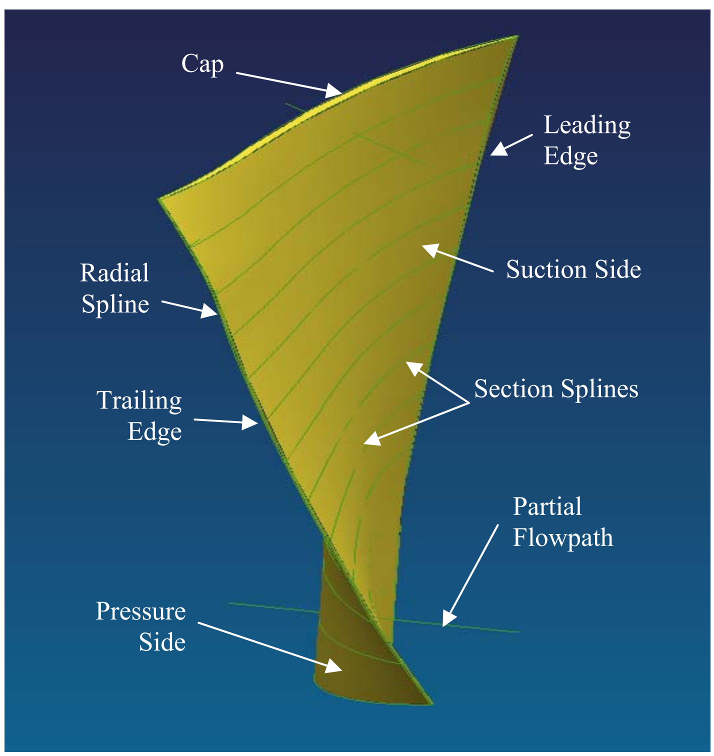


Figure 3: Airfoil model details

Although many other classes comprised the full compressor component, those discussed here represent the core functionality of the compressor utility. The turbine component used a similar set of classes to perform the same operation for constructing turbine geometry.

To simplify the use of the plug-in, a GUI was created using the NX User Interface Styler (UIS). Users opened the Airfoil Seed Part Generator dialog from a menu option or toolbar icon (see Fig. 4).

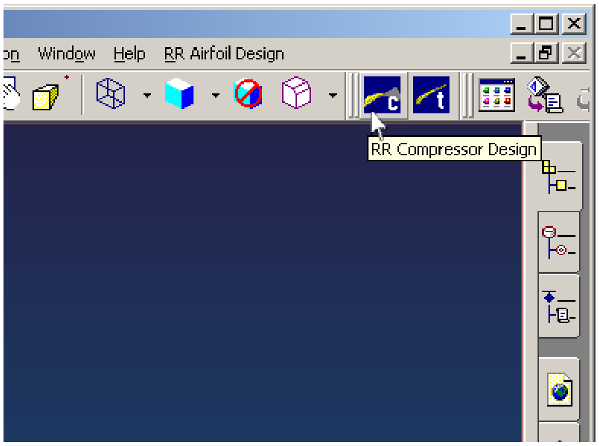


Figure 4: Application menu options and items

Once opened, the dialog gave users the options to select the airfoil definition file, select flow paths, and change modeling options and tolerances. The dialog also provided options to configure the continuity settings and layer color options (see Fig. 5). The turbine component GUI operated in a similar fashion, though it contained a number of features specific to turbine geometry.

After an airfoil file was generated, the user could then proceed to attach additional (stalk) geometry to it. The KF input parameters in the GUI remained associated to the geometry created by the plug-in. In this way, the airfoil geometry was now associative to the airfoil definition files. The ability of KF to create a dynamic model defined by rules made a plug-in based on this technology a viable solution for Rolls-Royce. When newer design revisions of the airfoil became available, the existing model could be regenerated to conform to the specifications of the new or updated definition file without requiring reconstruction of the attached geometry.

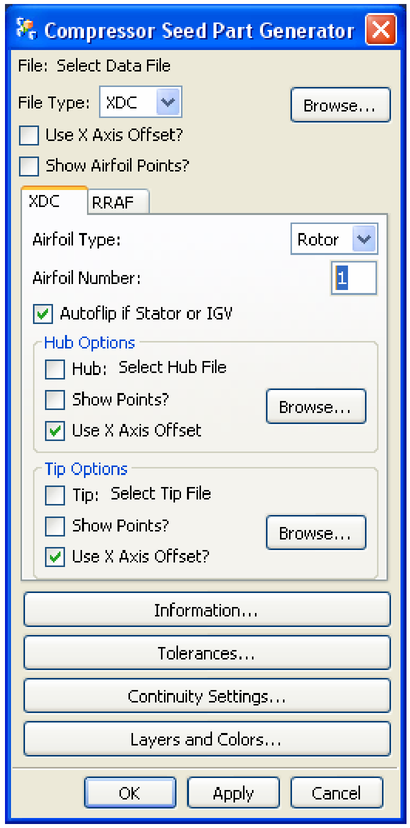


Figure 5: Application GUI

**Test Methodology**

In order to determine the value of the airfoil seed part generator, the product was tested based on the amount of time it could save over the current methods employed by Rolls-Royce. These methods created geometry that was not associative to airfoil engineering data, resulting in a static model that could not be easily updated to reflect changes in that data. With the associative geometry produced by the Airfoil Seed Part Generator, it was hypothesized that the use of the plug-in would yield significant time savings. Times to create various airfoils were obtained through the usage of a survey.

*Subjects*

Two expert participants were used in this study. Both were Rolls-Royce designers that were involved in the airfoil modeling process. These participants were expert users of NX and were identified as potential users of the Airfoil Seed Part Generator. Participant 1 has twenty-nine years of industry experience as a Designer in industry, with twenty-one years of experience at Rolls-Royce. He graduated from a technical institute with an Associate’s degree in Tool Design Technology and has been using 3D CAD tools since 1987. Participant 2 has similar experience. He is a Designer at Rolls-Royce, with twenty-two years of experience with the company. Similar to Participant 1, he has been using 3D CAD tools since 1987.

*Procedure*

Data on the previous methods to creating airfoils was obtained through experts in airfoil design at Rolls-Royce. These values measured the airfoil creation time from start to finish for a single iteration. The participants were shown an airfoil part and asked to estimate the time to model the geometry using traditional Rolls-Royce design methods. Once that time was recorded, they were then asked to use the Airfoil Seed Part Generator to create the given airfoil geometry. Participants were given a set of instructions to perform using the Airfoil Seed Part Generator while a researcher measured the length of time to complete the test. Participants opened the seed file and then launched the Airfoil Seed Part Generator. A text data file containing mathematical airfoil design data from the engineering department was loaded and all options to create the airfoil model to Rolls-Royce specification were selected. After the airfoil was generated, the researcher recorded the time to successfully generate an airfoil and any errors or shortcomings encountered in doing so. If the airfoil did not successfully generate on the first try, the time taken to edit the airfoil and create a usable model was recorded. Data was collected in this manner for thirty (n=30) airfoils. The time was recorded for each iteration, and a comparison was made between the iterations as shown in Figures 6 and 7. In order to provide the most conservative estimate of significance, values reported at 0 minutes were assumed to be 59 seconds and values reported at 1 minute were assumed to be 119 seconds (1 minute 59 seconds).

**Results**

Data obtained from experts at Rolls-Royce that estimated the time to create an airfoil iteration using current methods were found have a mean of $\bar{x} = 700$ seconds (11 minutes, 40 seconds) with a standard deviation of s = 458 seconds (7 minutes, 38 seconds). Time estimates ranged from 300 seconds (5 minutes) to 1200 seconds (20 minutes). Time estimates were used because Rolls-Royce did not keep a detailed record of airfoil creation time. Furthermore, exact times could not be recorded during the project due to time and monetary considerations.

Data measured to create the airfoil times using the Airfoil Seed Part Generator method ranged from 21 seconds to 442 seconds (7 minutes, 22 seconds). Lower times tended to appear when the model successfully generated on the first attempt and higher times appeared as outliers when the model failed to generate and a manual fix was applied. The mean for airfoil iteration creation time was found to be $\bar{x} = 81.93$ seconds with a standard deviation of s = 96.56 seconds.

It was hypothesized that the airfoil seed part generator would have a significant savings in time over the current methods in use at Rolls-Royce. To test the hypothesis ($H_a$: $\mu > \mu_0$), a one-sided t-test was used, resulting in values of t = 2.337. This result shows levels of significance at $\alpha = .05$ with a p-value between .02 and .01. These results are displayed in Table 1.

Table 1: Test Statistics

| | Current Method | Airfoil Seed Part Generator |
|---|---|---|
| Mean ($\bar{x}$) | 700 sec. | 81.93 sec. |
| Std. Dev. (s) | 458 sec. | 96.56 sec. |

t-value = 2.337 Alpha ($\alpha$) = .05 $.01 \leq$ p-value $\leq .02$

After removing outliers, the t-value increased to t = 2.432, however the p value remained between .01 and .02 for a highly significant result. This suggests that there is a significant savings in time by using the airfoil seed part generator as opposed to the current methods employed at Rolls-Royce. The average time savings of airfoil iteration remodeling, based on mean data from the control and test samples, was found to be 88.3%.

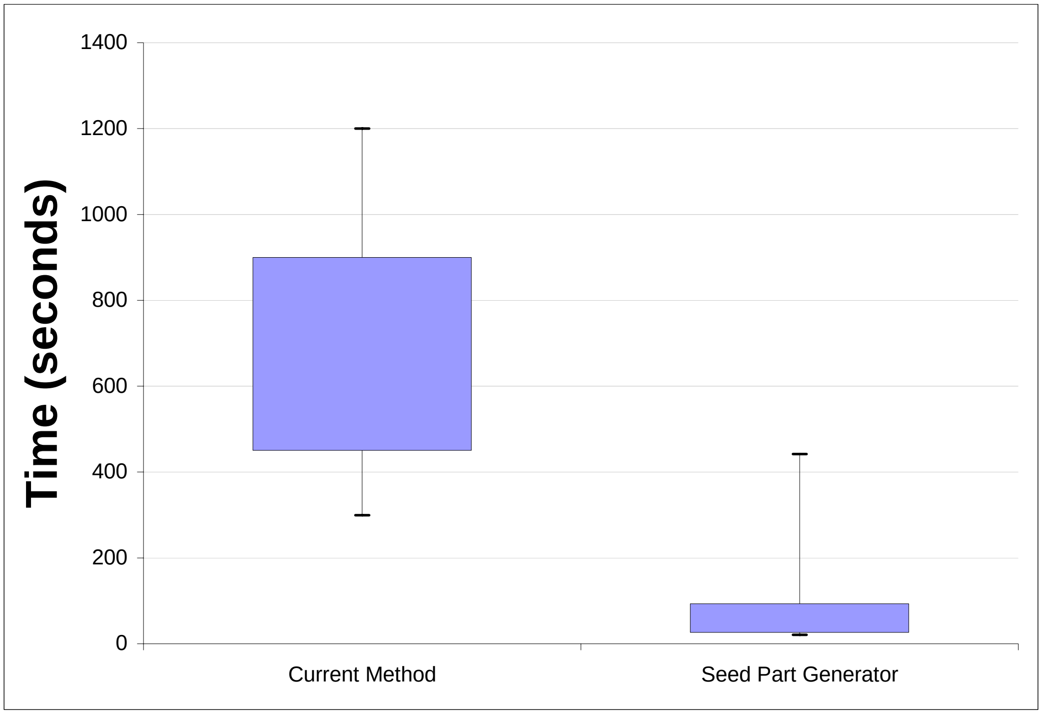


Figure 6: Box Plot of Test Data

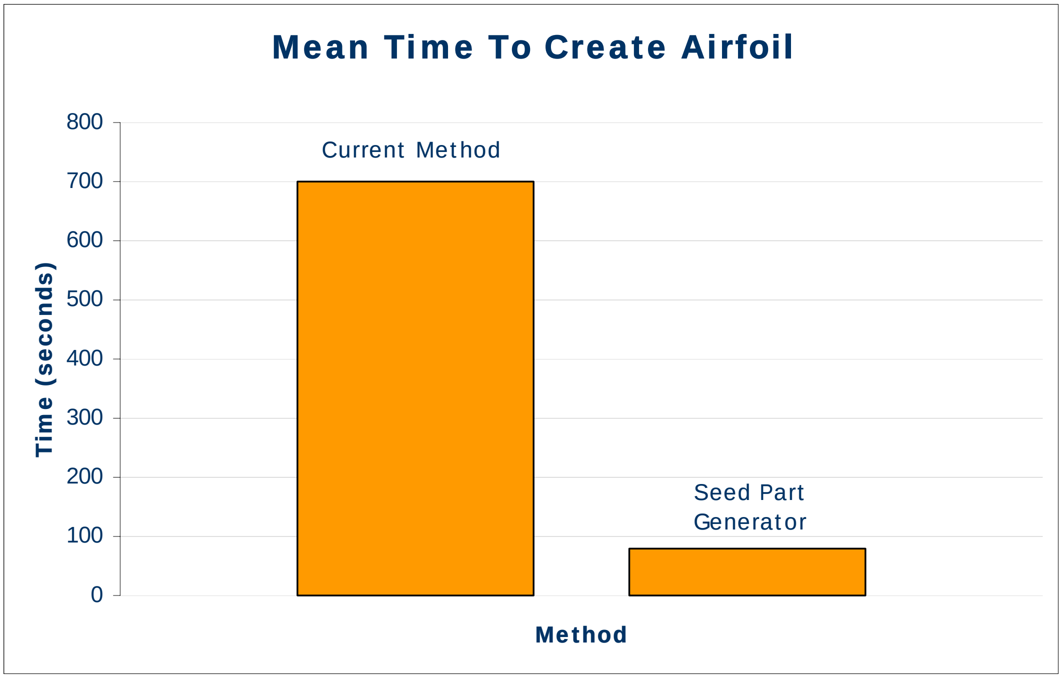


Figure 7: Comparison of Estimated Time Using Old Methods and Recorded Time with Seed Part Generator

## Discussion

The results of this test suggest a significant time savings for Rolls-Royce when using the airfoil seed part generator instead of the current methods. When the seed part generator failed to create geometry correctly the first time, it was usually due to a fillet feature not correctly forming. Making the model usable and generating usable geometry was easily done through a quick fix of removing the fillet, generating the next airfoil iteration and replacing the fillet manually. This same procedure is commonly done on models in the current Rolls-Royce airfoil modeling method. The occurrence of a non-generating iteration was relatively low (3 out of $n = 30$ samples). The variation in airfoil modeling time appears to have been reduced as well (Figures 6 and 7). However further research is needed to determine whether this is true. It should also be noted that because half ($n = 15$) of the values from the test data that were measured in minutes, these results were rounded up to the nearest minute to provide a conservative estimate. Even with these conservative estimates, very high levels of significance were still found. In actuality, the level of significance may be higher than the numbers reported.

In addition to the technical details of this project, curriculum implications for this project also exist. Current constraint-based CAD tools all create geometry in a very similar fashion: a profile is sketched on a 2D plane, dimensional and geometric constraints are added to the profile, and the 3D form is applied to finalize the geometry [18]. Given this process, the students were able to apply the basic modeling fundamentals and strategies they learned while using other CAD tools and quickly adapt to the structure and syntax of UG NX 2. By having been exposed to these modeling fundamentals in their course work, the students were able to quickly come up to speed on the NX 2 software used by Rolls-Royce. This is an example of the application of domain-specific knowledge in solving the problems encountered in the typical design process – the ability to apply strategic knowledge and usage of the CAD tool when engaged in a specific context [19, 20, 21]. By distilling the CAD tool into its common elements, it is not necessary for an academic institution to have the latest release of software, nor does it require them to have expensive maintenance. Three of the four students that worked on this project had never used NX 2 before this experience, but they were able to use existing knowledge of how CAD tools work in order to be knowledgeable of the software in a matter three weeks and complete the bulk of the project in a matter of about five weeks.

Anecdotally, this project was also beneficial to the students for gaining industrial experience on a project with a tight timeline and significant deliverables. As an added incentive, it was suggested at the beginning of the project that if the students met all of the major milestones of the project, they would be taken to Rolls-Royce's corporate office in the UK to present their results. While this was a good reward for the students for a job well done, it also exhibits the nature of a company that has invested itself in partnering with higher education to solve particular problems. It also demonstrated the benefits of academic / industry partnerships in an area that does not otherwise have many of these opportunities: CAD data management and engineering design graphics.

## Future Areas of Research

Several areas of the project have the potential to be further developed. The addition of several basic features to modify the created geometry such as the ability to shift the airfoil sections and the ability to tilt the airfoil would increase the value of the plug-in as a time-savings tool. These operations normally require new airfoil definitions and remodeling. Further development of the turbine core geometry would increase the associativity of this plug-in to the engineering data and increase the speed at which thermal engineering changes can be made.

Knowledge Fusion also has the ability to interact with TeamCenter, the UGS PLM/PDM software used at Rolls-Royce. Capability could be added to track design changes and read in previous designs from the database. PDM integration would allow for airfoil designs to be shared in real-time globally between all divisions of Rolls-Royce. Another area this plug-in could effect is the engineering optimization process which uses various computerized analyses to refine the airfoil design. This plug-in could allow batch analysis processing of these analyses.

## References

[1] Korakianitis, T., & Pantazopoulos G. I. (1993). Improved turbine-blade design techniques using 4$^{th}$-order parametric-spline segments. *Computer-Aided Design, Vol. 25*(5), 289-299.

[2] Corral, Roque, & Pastor, Guillermo. (2004). Parametric design of turbomachinery airfoils using highly differentiable splines. *Journal of Propulsion and Power, Vol. 20*(2), 335-343.

[3] Rolls-Royce Corporation. (2005, August). *NX2 knowledge fusion: Airfoil creation & morphing – Initial trial*. Indianapolis, IN: O'Brien, W., & Rasche J.

[4] Bermell, P. and Fan, I. (2002). *A KBE System for the Design of Wind Tunnel Models Using Reusable Knowledge Components*. Paper presented at the VI International Congress on Project Engineering. Barcelona, Spain.

[5] Prasad, Brian. (2005). What Distinguishes KBE from Automation. *COE NewsNet*. Retrieved April 27, 2006 from http://www.coe.org/newsnet/Jun05/knowledge.cfm#1.

[6] Rosenfeld, Lawrence. (1995). Solid modeling and knowledge-based engineering. In D. LaCourse (Eds.), *Handbook of Solid Modeling* (pp. 9.1-9.11). New York: McGraw-Hill.

[7] UGS Corporation. (2004). *Knowledge fusion for users: 2004 PLM World*. Retrieved on February 15, 2005 from pdf file.

[8] Hunter, R., Rios, J., Perez, J.M., & Vizan, A. (2005). A functional approach for the formalization of the fixture design process. *International Journal of Machine Tools & Manufacture, 46,* 683-697.

[9] Clark, A. L. (2001). A solid modeling services architecture for KBE applications. *ACM Symposium on Solid Modeling and Applications: Proceedings of the sixth ACM symposium on Solid Modeling and Applications.* Retrieved February 9, 2006, from ACM Portal database.

[10] Sekiya, T., Tsumaya, A., Tomiyama, T. (1998). Classification of knowledge for generating engineering models: A case study of model generation in finite element analysis. Research in Artifacts, Center for Engineering, University of Tokyo, Japan.

[11] Carleton, S. (2005). *Design rationale in a knowledge-based computer-aided design environment.* Unpublished master's thesis. Purdue University, West Lafayette, IN.

[12] Susca, L.[1], Mandorli, L.[2], Rizzi, C[1]. (2000). *How to represent intelligent components in a product model.* Department of Industrial Engineering, University of Parma[1], Italy. Department of Mechanical Engineering, University of Ancona[2], Italy.

[13] Park, J., Park, S., Hwang, I., Moon, J., Yoon, Y., Kim, S. (2004). *Optimal blade system design of a new concept VTOL vehicle using the Departmental Computing Grid system.* School of Aerospace and Mechanical Engineering, Seoul National University, Seoul, Korea.

[14] Spitz, W., Golaszewski, R., Berardino, F., Johnson, J. (2001). Development cycle time simulation for civil aircraft. *NASA Langley Technical Report Server*. Retrieved from ACM Portal database, February 16, 2006.

[15] Phillips, R. E. (1997). Dynamic objects for engineering automation. *Communications to the ACM, 40(5).* Retrieved February 9, 2006, from ACM Portal database.

[16] Kok, Raymond. (2004). *Knowledge Fusion Tips and Tricks*. UGS Corporation

[17] Collaborative Product Development Associates, LLC. (2004). *The intelligent master model process for achieving functional design.* Retrieved February 14, 2006, from http://www.cpd-associates.com.

[18] Wiebe, E.N. (1999). 3-D constraint-based modeling: Finding common themes. *The Engineering Design Graphics Journal, 63*(3), 15 - 31.

[19] Hartman, N.W. (2004). The development of expertise in the use of constraint-based CAD tools: Examining practicing professionals. *The Engineering Design Graphics Journal, 68*(2), 14-25.

[20] Hartman, N.W. (2005). Defining expertise in the use of constraint-based CAD tools by examining practicing professional. *The Engineering Design Graphics Journal, 69* (1), 6-15.

[21] Hartman, N.W. & Branoff, T.J. (2005). Learning theories: Applications for instruction in constraint-based solid modeling and other engineering graphics topics. *The Engineering Design Graphics Journal, 69* (2), 6-15.

**Biography**

William O'Brien is an undergraduate student in the Computer Graphics Technology program at Purdue University. He has been an intern with Roll-Royce.

Michael Chandonais graduated from the Computer Graphics Technology in the Spring 2006 semester. He is employed by Rand Worldwide as an Applications Specialist providing customer support in the use of CAD technologies.

R. Scott Henderson graduated from the Computer Graphics Technology in the Spring 2006 semester. He is employed by Rand Worldwide as a Technical Training Specialist providing customer training and technical education in the use of CAD technologies.

Mark Prezbindowski graduated from the Computer Graphics Technology in the Spring 2006 semester. He is employed in the interactive media field.

Nathan W. Hartman, Ed.D. is an assistant professor in the Computer Graphics Technology department at Purdue University. His research areas include the development and definition of expertise in the use of computer graphics tools, human spatial visualization abilities, and defining computer graphics as a discipline.

Joseph A. Rasche is a Senior Engineer and Specialist for Rolls-Royce Corporation in Indianapolis. His area of expertise is in structural analysis, and he also serves on the corporate Engineering Methods group.

## Glossary

**.dfa** – (Design File ASCII) contains Knowledge Fusion classes and scripts

**.dlg** – (dialog) UI Styler components

**Airfoil** – generic term for a blade used on or in an aircraft engine

**Airfoil Definition** – text file which lists all the points associated with a specific airfoil design

**Blade Book** – airfoil definition file format used by Rolls-Royce U.S. turbine engineering department

**Computational Fluid Dynamics (CFD**) – an engineering analysis tool that computes the movement of a fluid (gas or liquid) as it interacts with a system

**Compressor Blade** – thinner airfoil used to compress and guide air at the front of a turbine engine.

**Engine Axis** – the central axis of a turbine engine running along its length

**Finite Element Analysis (FEA)** – An engineering analysis tool that breaks a computer model into many simply shaped finite elements. Mathematical equations can be run on the finite elements for a total analysis of the entire object.

**Flowpath** – The path air moves through an engine along the engine axis. Bounded at the hub and tip.

**Hub** – The section of the airfoil nearest to the engine axis

**Inner Guide Vane (IGV)** – airfoils which help to guide air into the compressor

**Knowledge Fusion (KF)** – "an interpreted object-oriented language that has been designed to permit an end-user to easily add engineering knowledge to the task at hand..." (UGS NX3.0 documentation)

**Leading Edge** – edge that leads the airfoil through air (see Fig. 6).

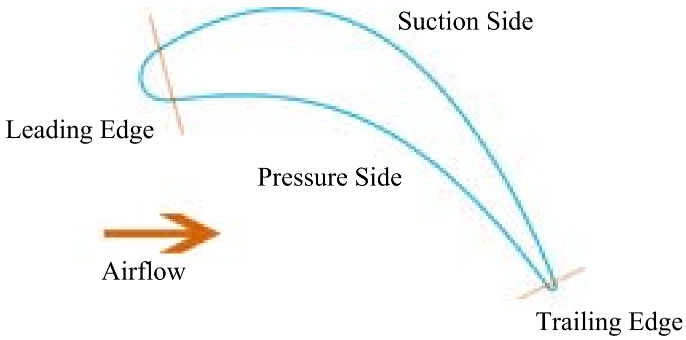


Figure 6. Compressor Airfoil Section Diagram

**NX2/3** – versions of the Unigraphics CAD package used throughout the project.

**Pressure Side** – the side of the airfoil that creates pressure as it rotates (see Fig. 6)

**Radial Axis/Direction** – the axis perpendicular to the engine axis

**RRAF (Rolls-Royce Airfoil)** – a Rolls-Royce airfoil definition file format developed in the United Kingdom

**Rotor (blade)** – rotating airfoil to either compress air (compressor) or be rotated by air (turbine).

**Seed Part** – a modeling part file with basic settings, information and possibly standard datums or geometry already created to be used to start modeling a new part

**Stator (blade)** – static airfoil guides and controls airflow between rotors.

**Stack** – the position of airfoil sections along the radial direction to define the airfoil

**Stalk** – geometry that connects the airfoil to the disk.

**Section** – an outline of an area of an airfoil. Sections may be the result of planar or conical intersection of an airfoil. Sections are stacked along the radial direction.

**Suction Side** – the area of airfoil opposite the pressure side. As this part rotates through the air it leaves a gap creating suction. (see Fig. 6)

**Tangency Points** – the points on a section where the sides meet the edges (four per section)

**Thermal Barrier Coat (TBC)** – coating applied to the exterior of airfoils to prevent overheating.

**Tip** – the section of an airfoil farthest from the engine axis

**Turbine** – a fan like structure that is driven by a fluid

**Turbine Blade** – an airfoil on a turbine

**Turbine Engine** – An engine wherein compressed air is mixed with fuel and ignited to drive one or more turbines. The turbine drives a shaft that powers the compressor and possibly a fan, propeller or generator.

**Trailing Edge** – last section of an airfoil to move through the air as it is rotated (see Fig. 6)

**XDC** – airfoil definition file format used by Rolls-Royce U.S. compressor engineering department.