*Computer-Aided Design and Applications*

http://www.cadanda.com

# CAD-Centric Dynamic Workflow Generation

Travis L. Kenworthy[1] and C. Greg Jensen[2]

[1]Brigham Young University, travisk@byu.net
[2]Brigham Young University, cjensen@byu.edu

**ABSTRACT**

An automated method has been created to perform optimization iterations on CAD models utilizing CAD external analysis tools. The method has been implemented in a program, called the Process Integrator, which is launched from within a CAD system and automatically generates and executes optimization workflows in the software package iSIGHT-FD by Engineous. Test cases are presented in which the efficiency of the automated process is demonstrated.

**Keywords:** CAD, optimization, automation, workflow, iSIGHT-FD.
**DOI:** 10.3722/cadaps.2009.673-683

## 1. INTRODUCTION

The use and automation of CAD tools can enable a company to have a competitive advantage and improve market share by speeding up the design process and increasing productivity. CAD models are important design tools that enable the designer to conceptualize, visualize, and analyze a design. Parametric CAD modeling allows model features to be easily manipulated through the use of expressions associated with the feature parameters [1]. The use of parametrics in modeling creates the ability to quickly perform design iterations on CAD models and opens the doors to more frequent usage of CAE design tools [3]. Additionally, high-end CAD packages, such as NX by Siemens, CATIA by Dassault Systemes, and Pro-E by PTC, provide several built-in design tools, including finite element analysis, dynamic analysis, and many other useful tools that can help a mechanical designer to refine designs and leverage engineering knowledge in CAD models. Despite the existence of the many CAD internal applications, designers and engineers in industry often use various proprietary, custom non-CAD methods or standalone external analytical processes that constrain or evaluate a design [8]. Unfortunately, there is often a disconnect between the part definition contained in CAD and analyses external of CAD. An efficient method is needed to obtain part definition from CAD models, run analyses using the data, and modify the CAD models to reflect the optimal results found from analysis iterations. This need has been met by creating a method and program prototype that seamlessly integrates external methods and analytical tools with interactive CAD sessions.

### 1.1 Current CAD Automation

The simplest, non-automated method to obtain part definition information from a CAD model is to manually extract the desired part parameters by looking them up in the model. This method is very time consuming and is not a practical solution for performing many design iterations. Lamarche took full advantage of the benefits of automating CAD processes. Through automating modeling tasks using a CAD system's Application Programming Interface (API), a 320 minute modeling task was reduced to a 4.7 minute, fully automated process [6]. Using similar automation techniques, Dye reports of the development of a gas turbine design tool, called Cross-Section Designer, which provides the user with an editable engine gas path and allows the user to manipulate geometry in various design schemes [2]. The Cross-Section Designer is an example of how engineering knowledge can be merged with automation techniques to control a design. The concept of knowledge based automation is critical for creating a competitive advantage by not only automating modeler tasks, but by realizing engineering knowledge in the designs. An additional concern in this area is that increasing the automation of a process can decrease the ability to innovate within that process [10]. This problem can be minimized by creating custom applications that are generic and adaptable to changes in the process.

### 1.2 Current CAD-Analysis Integration

Several researchers have been working on problems related to integrating analytical systems with CAD [7]. Kosavinta describes the integration of CAD with a Decision Support System (DSS) with the goal of enabling the designer to make real-time decisions about a design. The DSS evaluates the project, calculating project feasibility and cost, enabling the designer to quickly make decisions about a project design [5]. Similarly, Shehab details the integration of a knowledge based system (kbs) with a CAD system that has the capability to help the designer create an optimal assembly based on estimated assembly times and cost, as well as providing design improvement suggestions [11]. King utilized the API interfaces of Unigraphics (now called NX), Hypermesh, and Fluent to create a CAD-centric approach to CFD analysis [4]. These examples are the types of processes that a generic process integrator could aid in incorporating with a CAD system. The integration method that was performed in these examples created a program specifically to link a CAD package to the desired analysis. This method was efficient for executing the specific analysis or application, but would be repetitive for linking multiple analysis applications with the CAD system because it requires that a new application be written to incorporate each analytical tool. A more generic approach to process integration would enable a designer to interact with many CAD-external applications through a single custom tool.

### 1.3 Current CAD Optimization

A method to incorporate analytical tools with CAD has been created by Engineous using their optimization software package iSIGHT-FD. According to Velden, Engineous has created an automated process to drive geometric changes in CAD models using components in iSIGHT-FD [12]. This methodology presents a flexible approach to CAD/CAE integration because it allows CAD models to be integrated with any combination of analyses. However, this method has the limitation of requiring the manual configuration of each component to be used, which makes the optimization of many individual parts from an assembly very time consuming. Similar to the automation of repetitive modeling tasks in CAD systems, an automated method to apply a generic optimization configuration to many individual parts can provide significant time savings as well as make otherwise impractical goals of optimizing hundreds of individual parts with similar but different optimization configurations feasible. Additionally, an advantage of automating the task of creating optimization workflows for CAD models is the possibility of expanding the user base for using engineering analysis and optimization tools by creating a method for CAD modelers to incorporate analysis results into their models that doesn't require knowledge of optimization packages. An enhancement to this method would be to create a program that can be launched from within a CAD system, automatically create iSIGHT-FD optimization workflows, run optimization iterations, and use the results to update the current CAD model. The benefit of this enhancement is the increased amount of automation in the process, saving time in both the set-up and execution as well as minimizing the needed training of the user to use optimization and analysis software.

## 2. PROCESS INTEGRATOR IMPLEMENTATION

The approach taken to merge CAD with external CAE tools was to concentrate efforts on making a process that is catered to CAD modelers/design engineers, resulting in a process that is centered on the CAD environment. A method has been devised in which a program that resides within the CAD environment can allow the user to configure and run a process with all the user interaction taking place within the CAD environment, see Fig. 1. When a process is run, the program utilizes an optimization package to create workflows based on the current process configuration. The workflows make part iterations and link them to analysis for optimization without the user needing to know how to manage an optimization environment or the precise details of the analysis. After the optimization has been run, the program displays results in the CAD environment. This method has been devised with the intent to minimize the training required to allow CAD modelers to leverage engineering knowledge in their models through the implementation of optimization methods in conjunction with part analyses.

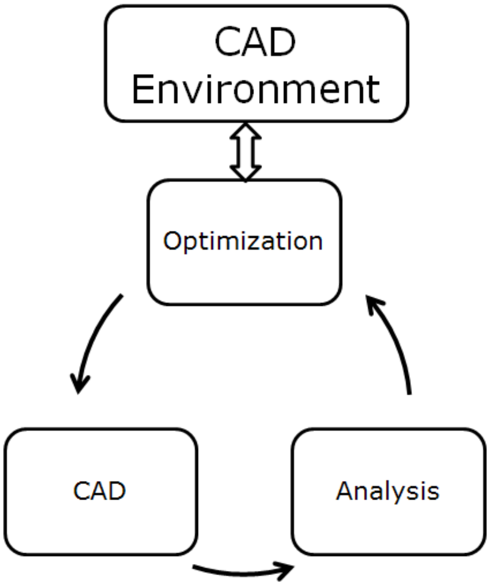


Fig. 1: Process flows from CAD environment to optimization package then back to CAD.

**2.1 Program Layout and Communication**

The program created to implement the method discussed, called the Process Integrator, can be segmented into the following applications that work together to perform the required tasks of seamlessly merging analytical processes with CAD models.

- Process Integrator—CAD Internal Application
- Process Initializer—CAD Internal Application
- iSIGHT-FD Launcher—Windows Batch File
- iSIGHT-FD Workflow Builder—iSIGHT-FD Application
- NX Component—iSIGHT-FD Component Application/CAD Batch Application

The program has been implemented using the Siemens NX4 CAD package, the Engineous iSIGHT-FD optimization package, and a Windows batch file. Application programming interfaces of these packages have been used via the JAVA programming language to create an automated process. Communication between the two programs is done with text files that convey information back and forth between the different programs. Each of the program segments will be discussed in some detail.

**2.2 Process Integrator**

The Process Integrator is a custom application developed to run in Siemens NX4, widely known as Unigraphics. The program is designed for the user to run with a part or assembly open that they would like to run analysis iterations on. The program is run by pressing a button on the NX toolbar, as seen in Fig. 2.

When a user runs one of the available processes, the program reads a saved initialization file for that process and examines it against the current CAD assembly to see if the process applies to any of the parts. All of the data needed to generate optimization workflows for the parts in the assembly is gathered and saved in a file to be read by the iSIGHT-FD Workflow Builder.

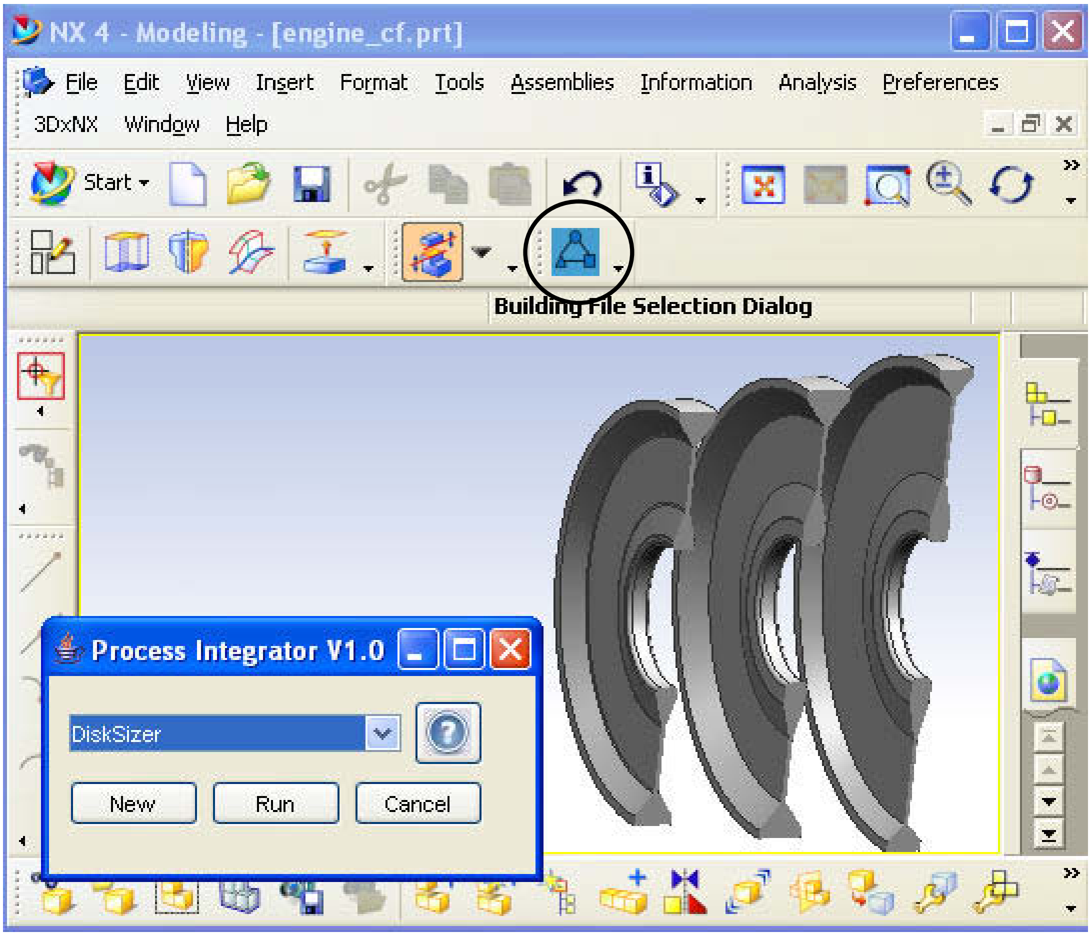


Fig. 2: Process Integrator tool made accessible from NX toolbar (button circled).

Once the data required to run the part optimizations on the assembly have been acquired and saved for later use, a batch file called the iSIGHT-FD Launcher is run. The iSIGHT-FD Launcher sets various environment variables needed by iSIGHT-FD and then launches iSIGHT-FD while indicating a custom application, called the iSIGHT-FD Workflow Builder and described in Section 2.4, to be run. The iSIGHT-FD Workflow Builder creates and executes optimization workflows and saves the results in a text file for the Process Integrator to read later.

The Process Integrator waits for the iSIGHT-FD optimizations to be completed and then reads the results file created by the iSIGHT-FD Workflow Builder. The resulting optimal design variables, objectives, and constraints are then listed for the user in NX4 as well as realized in the current CAD models.

One of the most important objectives of the Process Integrator is that it be used to integrate many different types of analysis with CAD assemblies. This objective was met by the creation of the Process Initializer which allows for the creation of generic process configurations.

**2.3 Process Initializer**

The Process Initializer is a graphical interface that makes it possible for the user to configure an optimization that will link CAD models to processes that can be called from a command line and that read inputs and write outputs from text files. The Process Initializer, as displayed in Fig. 3, has five tabs that enable the user to configure a process in an intuitive fashion.

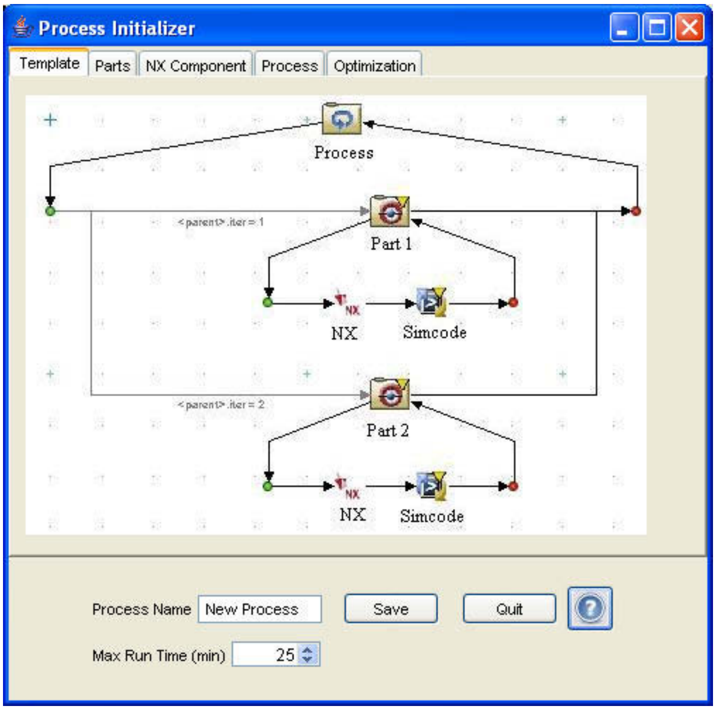


Fig. 3: Process Initializer Template tab.

*2.3.1 Template Tab*

The first tab of the Process Initializer displays an example iSIGHT-FD workflow template. This template demonstrates the layout that will be created in iSIGHT-FD based on the configuration created in the Process Initializer. The next four tabs are used to provide the information to configure each of the components in this template. The top of the workflow shows a Loop Component, identifiable by a circular arrow, which drives the entire workflow by iterating one at a time through each of the part optimizations to be run. The number of parts to be optimized is determined by the information provided in the Parts tab as described in Section 2.3.2. Each part optimization is controlled by an Optimization Component, identifiable by a bulls-eye, that handles iterations of design variables to meet design constraints and objectives. Each optimization loop also contains an NX and Simcode Component. The NX Component handles NX part updates and data extraction and is configured on its corresponding tab. The Simcode Component handles updating an input file for an analysis to read, executing the analysis, and reading the analysis output file.

*2.3.2 Parts Tab*

The Parts tab, as seen in Fig. 4, recreates the current NX part tree and allows the user to specify which parts to be optimized for a given process. The Parts tab allows the user to choose one of three possible part configurations, including Part List, All Parts, or Subset of Parts. The Part List configuration allows the user to select any number of parts from the Available Parts List to optimize. The All parts configuration will configure the process to optimize all the parts in the assembly. Finally, the Subset of Parts configuration allows the user to specify a set of parts based on a specified naming convention. For example, if the part names are specified to contain the character string “disk” then all parts in the assembly with that character string will be optimized. The Subset of Parts configuration is a favored configuration because once the process configuration is saved it can be run on any part assembly and if the parts are named with a specific convention then the process configuration can be used to optimize a desired set of parts regardless of the full part names.

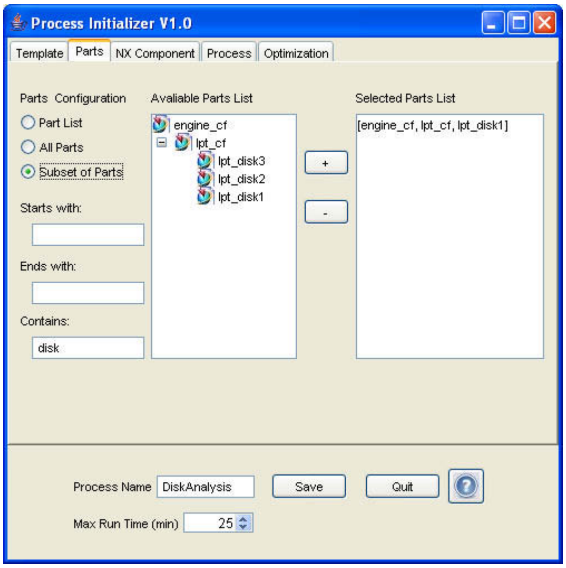


Fig. 4: Process Initializer Parts tab.

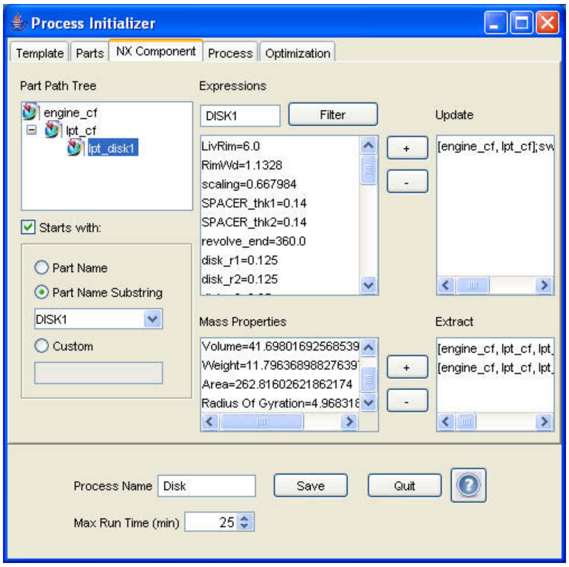


Fig 5: Process Initializer NX Component tab.

### *2.3.3 NX Component Tab*

The NX Component tab, as seen in Fig. 5, allows the user to specify which part expressions to update to perform design iterations, as well as which expressions and mass properties to extract from the assembly for use in analyzing and evaluating the assembly. The Part Path Tree allows the user to specify which part in the assembly to update or extract expressions from and the relative part locations are used to update/extract expressions from any needed part based on the part to be optimized. Additionally, the "Starts with:" checkbox allows the user to specify how an expression name might change in relation to the part being optimized. For example, if lpt_disk1 is configured to be optimized by changing the expression DISK1_width, then lpt_disk2 would automatically be optimized by changing DISK2_width.

### *2.3.4 Process Tab*

The Process tab, as seen in Fig. 6, is used to define an analysis to link to the CAD models. This tab allows the user to specify any set of command line calls or arguments. Additionally, input and output text files for the analysis can be specified. Specifying the input and output files that the analysis will use allows iSIGHT-FD to update the analysis inputs as design variables or output parameters from NX and to read the outputs and use them as design constraints or objectives.

### *2.3.5 Optimization Tab*

The Optimization tab, as seen in Fig. 7, allows the user to specify which of the parameters from the NX and Simcode Components will be design variables, design constraints, or design objectives. Additionally, the user can specify bounds and weights for the different parameters and optimization directions for the objectives.

## **2.4 iSIGHT-FD Workflow Builder**

The iSIGHT-FD Workflow Builder is a program that runs in iSIGHT-FD and builds an optimization workflow, following the template displayed in Fig. 3, and as defined by the configuration created by the Process Initializer. The Workflow Builder program performs four basic steps that will be described.

1. Create connection to iSIGHT-FD
2. Build the iSIGHT-FD optimization workflow
3. Run the iSIGHT-FD optimization workflow
4. Retrieve Job results from iSIGHT-FD

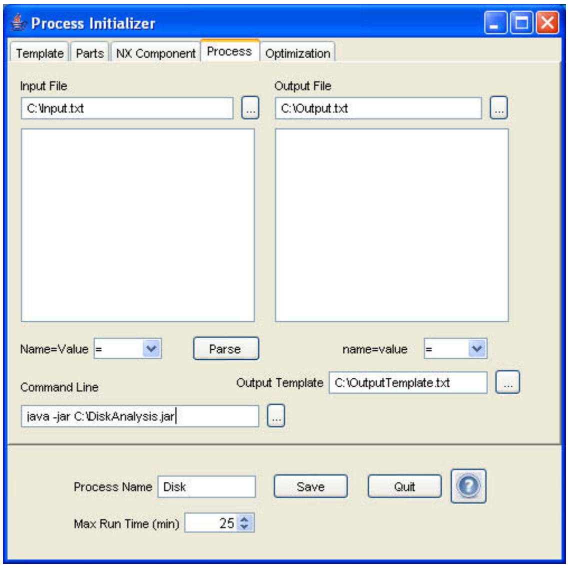


Fig. 6: Process Initializer Process tab.

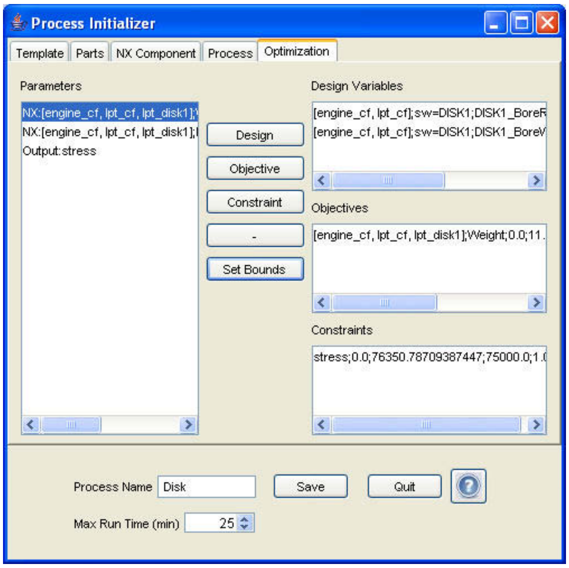


Fig. 7: Process Initializer Optimization tab.

The connection to iSIGHT-FD is made using the iSIGHT-FD Software Development Kit which allows access to many iSIGHT-FD functions. The iSIGHT-FD Workflow Builder creates a workflow driven by a Loop component that iterates through the optimization sub-loops to optimize each of the parts. Each optimization component is configured to utilize the design variables, constraints, and objectives that correspond to the NX part to be optimized by its sub-flow. Each NX Component is configured to update or extract different expressions and mass properties depending on the part to be optimized. All of the Simcode Components will be configured to run the same analysis, but each Simcode Component is configured to have its parameters mapped to the corresponding NX parameters for its sub-flow.

After the workflow has been created, an iSIGHT-FD Job is created to run the workflow. While the Job is running the iSIGHT-FD Workflow Builder makes queries to the run engine at regular time intervals to check the progress and indicate it to the user. When the Job is completed, the results from each optimization loop are retrieved and saved in a text file for later use by the Process Integrator.

### 2.5 NX Component

The NX Component was created using both the iSIGHT-FD Software Development Kit and the NXOPEN application programming interface allowing it access to both functions from iSIGHT-FD as well as NX. The NX Component is able to take property and parameter values from the iSIGHT-FD environment and use them to execute functionality in NX, such as opening a part assembly and updating expression values. Additionally, the component is able to obtain NX part expression and mass property values and use them to update iSIGHT-FD parameter values. Another important part of the functionality of the NX Component is an initialization function that obtains an NX Session and holds onto it for the duration of the iSIGHT-FD session, meaning that the time intensive process of acquiring the NX session only needs to take place once.

## 2.6 Disk Analysis Test Case

A disk analysis test case was performed in which the disks in a jet turbine engine assembly were optimized. Several dimensions of the disk were used as design variables and the assembly was linked to an analysis to run a stress calculation based on part expressions and mass properties. Maximum stress was used as a design constraint. Disk weight was minimized as the design objective. Two methods were used to perform the test case. The first method was a base test in which iSIGHT-FD was used manually to create the workflow using a custom NX Component as the only custom technology. The second test was performed using the Process Integrator program to create a process configuration that automatically creates the workflow. The two methods were used to create the exact same iSIGHT-FD workflow, as seen in Fig. 8. Since both methods create the same workflow the time to configure the workflow and not execution time will be examined.

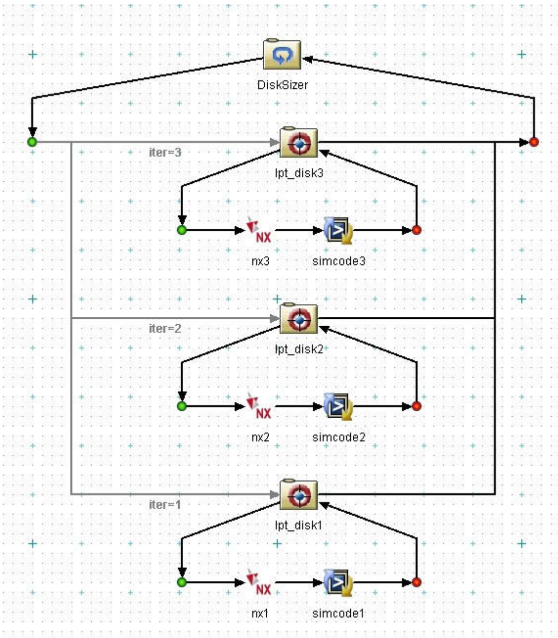


Fig. 8: Optimization loop used to optimize 3 turbine disks.

Both methods were performed by an experienced user of both the Process Integrator and iSIGHT-FD, and the resulting times to configure an optimization loop for each disk in the assembly are displayed in Tab. 1. The results of the tests indicate that the workflow created using the Process Integrator took approximately 2/3 of the time to create as to manually create a single optimization loop. Once a generic process configuration has been defined, the Process Integrator takes approximately 3 seconds to create each loop. Manually, after the first optimization loop was created, the subsequent loops were created by creating a copy of the first loop and modifying the names of the expressions and parts to be used in the optimization.

| Part # | Manual (min:sec) | Process Integrator (min:sec) |
|---|---|---|
| 1 | 18:27 | 12:48 |
| 2 | 3:15 | :03 |
| 3 | 3:10 | :03 |
| Total | 24:52 | 12:54 |

Tab. 1: Time to configure an optimization with one optimization loop per part.

An extrapolation of the test data, as seen in Fig. 9, demonstrates how the time to create a workflow with many parts to be optimized stays relatively constant using the automated process, while the time to create them using manual methods grows to undesirable quantities.

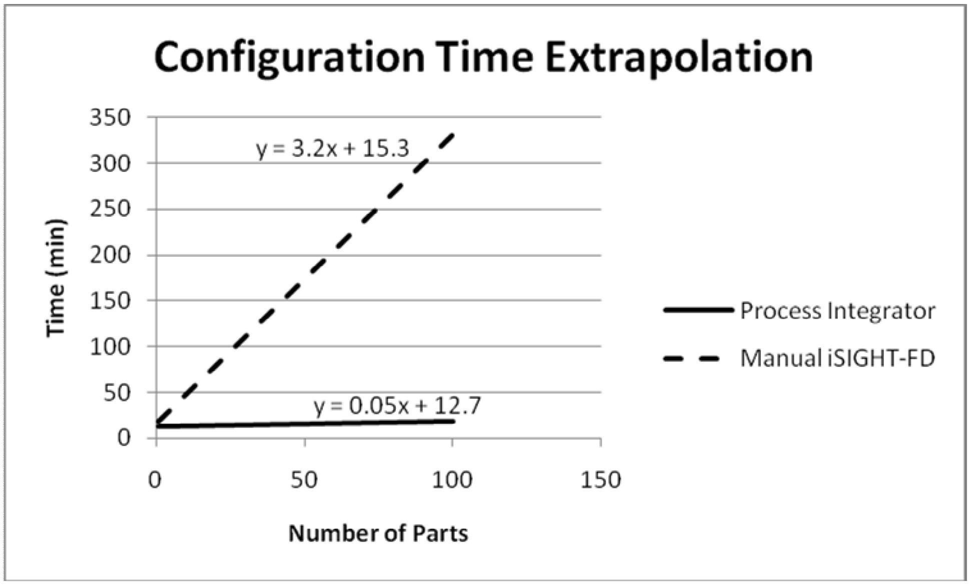


Fig. 9: A time extrapolation indicates the benefits of the automated process.

The benefits of using the Process Integrator become evident when it is desired to optimize many individual parts in an assembly. The Process Integrator is able to use a generic process configuration to identify parts with a given naming convention and use them to automatically create a workflow with many parts to be optimized that would take a great deal of time to do manually.

**2.7 Static 25-Bar Truss Test Case**

A static 25-bar truss optimization, as displayed in Fig. 10, was performed to demonstrate the capabilities of this method to configure and execute a complex system optimization. A parametric CAD model of the truss was created and the cross-sectional areas of each bar in the truss were used as design variables. The objective of the design problem was to minimize weight with a maximum stress constraint. The Process Integrator was used to link the CAD model to an analysis performed in MATLAB to calculate the stress in each bar of the truss [9].

The truss optimization was easily configured using the Process Integrator. The optimization automatically performed 1857 design iterations during 3.5 hours to find the combiniation of bar sizes with the minimal weight that meets the maximum stress constraint.

**3. CONCLUSION**

The major advantage of the Process Integrator over conventional methods, as demonstrated by the test case, is the ability to store a generic process configuration that can be selected and run on any assembly and that will automatically find all the parts that apply to that process and the corresponding parameters needed to create the optimization for those parts. One of the shortcomings of the Process Integrator is that the pre-defined workflow template is restrictive as it allows the NX assembly to be connected to only a single analysis in a single configuration layout. Regardless, the Process Integrator creates a huge advantage for creating simple workflows that link CAD models to a single analysis at a time because of the fact that it can automatically create a virtually unlimited number of optimization loops. Additionally, the Process Integrator reduces the amount of training required to enable mechanical designers to run analysis on CAD models. The Process Integrator is a step forward towards building engineering knowledge into CAD models, which can lead to more accurate models at an earlier design stage and to big savings in time and money throughout the design cycle.

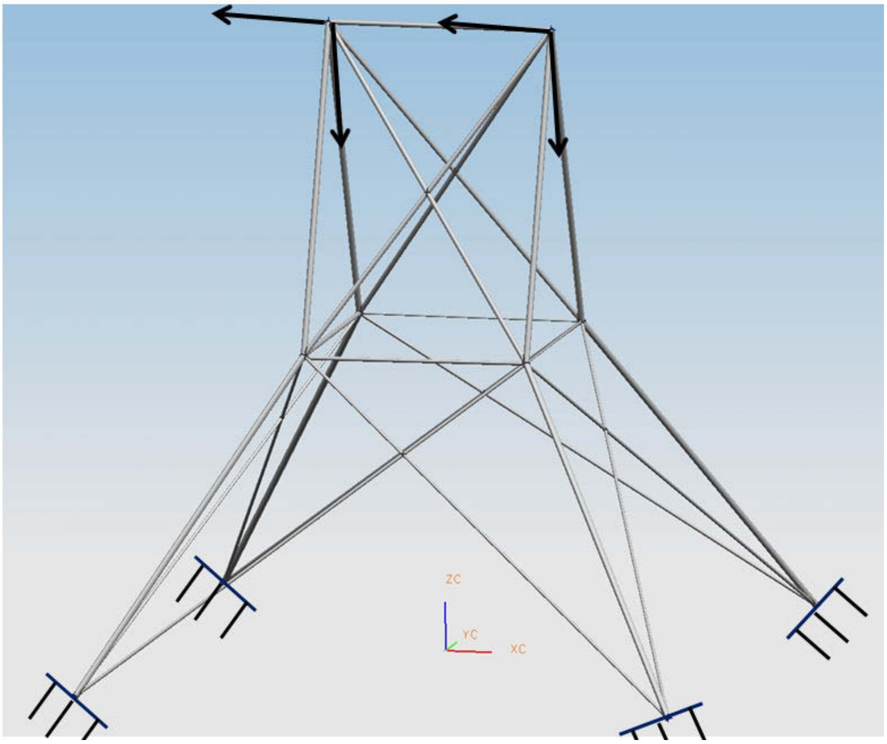


Fig. 10: Static 25-bar truss test case.

**4. ACKNOWLEDGEMENTS**
We wish to acknowledge and thank the great people and Pratt & Whitney for sponsoring this research.

**5. REFERENCES**
[1] Berglund, C. L.; Jensen, C. G.: Robust Parameterization Schema for CAx Master Models, Computer-Aided Design & Applications, 5(5), 2008, 715-729.
[2] Dye, C.; Staubach, J. B.; Emmerson, D.; Jensen, C. G.: CAD-Based Paramertric Cross-Section Designer for Gas Turbine Engine MDO Applications, Computer-Aided Design & Applications, 4(1-4), 2007, 509-518.
[3] Elliot, J. H.; Berglund, C. L.; Jensen, C. G.: An Automated Approach to Feature-Based Design for Reusable Parameter-Rich Surface Models, Computer-Aided Design & Applications, 4(1-4), 2007, 498-507.
[4] King, M. L.: A CAD-Centric Approach to CFD Analysis with Discrete Features, Brigham Young University, 2004.
[5] Kosavinta, S.; Kanongchaiyos, P.; Jinuntuya, P.; Integration of CAD Software with DSS for Engineering and Architectural Project Design, Computer-Aided Design & Applications, 4(1-4), 2007, 467-476.
[6] Lamarche, B.; Rivest, L.: Dynamic Product Modeling with Inter-Features Associations: Comparing Customization and Automation, Computer-Aided Design & Applications, 4(6), 2007, 877-886.
[7] Lee, S. H.: A CAD-CAE integration approach using feature-based multi-resolution and multi-abstraction modeling techniques, Computer-Aided Design, 37, 2005, 941-955.
[8] Perlak, J.: Personal interview, Pratt & Whitney, November 2007.
[9] Rahami, H.: Truss Analysis, MathWorks, 2007,

http://www.mathworks.com/matlabcentral/fileexchange/14313.
[10] Salzman, H.: Computer-Aided Design: Limitations in Automating Design and Drafting, IEEE Transactions on Engineering Management, 36(4), 1989, 252-261.
[11] Shehab, E. M.; Abdalla, H. S.: A cost-effective knowledge-based reasoning system for design for automation, Proc. IMechE, 220(Part B: J. Engineering Manufacture), 2006, 729-743.
[12] Van der Velden, A.: CAD to CAE process automation through iSIGHT-FD, Proceedings of the ASME Turbo, Expo, 1, 2007, 87-93