Integration of Secondary Air System for Multidisciplinary Design Optimization of Gas Turbines

PEOC'H Timothé*, BAYOUMY Ahmed**, STANISZEWSKI Martin***, MOUSTAPHA Hany*, KOKKOLARAS Michael**, GARNIER François*

*École de Technologie Supérieure **McGill University ***Siemens Canada Limited

Keywords: Aero derivative engines, Gas turbines, Automation, Secondary Air System, Industry4.0

Abstract

This article presents the integration of Secondary Air Systems (SAS) tools into a Design & Analysis (D&A) platform developed in a Multidisciplinary Design Optimization (MDO) context.

As technology increasingly requires high precision, and therefore meticulous work in multi-expertise fields, gas turbine engineers and analysts suffer from non-value added tasks. Among these tasks stand data management, poor data transmission between software, and fastidious pre and post-processing; all of which reduce analysis time greatly and consequently the quality of the final product.

The objective of such a platform is to regroup all gas turbine software into a single platform to allow for automation. Therefore, tools are run in batch mode and the platform is linked to a data management system improving efficiency tremendously.

This platform has been designed and tested for SAS engineers. Through careful analysis of the workflow, a list of eligible tasks for (semi-)automation was established and prioritized. Pre-processing was semi-automated. A proposed setup based on input data greatly reduces human error. The process itself was fully automated allowing an entire power-curve calculation in a single run. Lastly, the final source of improvement was accomplished in post-processing. SAS tools produce hundreds of thousands of data values, as a means to simplify this readout, a task-depending synthesis page displaying parameters of interest has been created. A graph plotting tool has also been implemented reducing manual plotting on spreadsheets.

The desirable amounts of design iterations do not occur because of the complex and lengthy calculations. In order to perform iterations and later MDO, the SAS module integrates a feedback

loop to the performance module with the integration of SAS bleeds into the performance model to ensure model alignments.

The aforementioned functionalities have shown great potential. Thus far, non-value added tasks have been drastically reduced; consequently, time for analysis has been lengthened. In addition, result accuracy has greatly improved thanks to multi-iteration.

1-Introduction

The design complexity of gas turbines design has increased with the demanding market over the last decade. More specifically, fuel efficiency, pollution, and noise reduction are major contributing factors. Engineers are facing unprecedented challenges in order to provide products with improved lifetimes and that are safer, cleaner, more reliable, and more powerful.

The gas turbine market, much like other markets, is being stressed with short developing times. As a consequence, gas turbine manufacturers cannot afford to run long and complex iterative processes. However as historical companies' best practices struggle to fulfill new market requirements MDO is being widely developed. To address the issue of time, workflow automation is mandatory. However, if automation is occurring it must be integrated wisely so as to not limit engineering creativity [1]. This improvement will not only produce time benefits but also increase the design space exploration. All of these advantages ultimately lead to increased result accuracy.

Error reduction, requires that design tools become increasingly performant. One way this can be achieved is with Product Life Management (PLM), which also promotes a better collaboration of the workforce [2]. Unfortunately, there is currently no platform that allows for improved speed, accuracy and the collaborative efforts from different disciplines for aero derivative gas turbines. This project describes the integration of the Secondary Air System (SAS) in such a platform. This was accomplished by integrating gas turbines SAS design workflow within a single Multidisciplinary Design Optimization (MDO) platform in combination with PLM software. The outcome of this this effort was ultimately enhancing engineering efficiency.

2-Design & Analysis Platform

Framework

The current work is based on the framework developed by *Ramamurthy et al.* [3] in which they describe an object-oriented software allowing advanced data management and workflow simulation. In order to shorten the development cycle, they outline a framework which includes a Systems Engineering Module (SEM) that allows the user to select, configure, and execute a model after having linked all the required inputs. An Advanced Visualization Module (AVM) enhances the post-processing phase of the model execution by allowing the user to analyse results within the same tool as for the execution.

Their framework is python object-oriented based and provides an Application Programming Interface (API) allowing on-site developers to easily design new streamline workflows upon engineer's request. These workflows are displayed in a user-friendly user-interface (UI), "[minimizing] the engineers' necessity to get familiar with new concepts and interfaces".

Data management

The Design & Analysis platform will be used to integrate a powerful Product Life Management (PLM) tool; providing model and data management. This was necessary as keeping consistent data is required to reduce loss of time and errors.

Indeed, because of its design complexity, development teams for gas turbines are composed of many experts with a wide range of expertise. Each contributor is required to use their requirements and inputs to execute a model and analyse the results which will subsequently be passed on to other experts. This iterative process can be long and produces a significant amount of data. As a consequence, engineers are spending a significant amount of time finding up-to-date models or input files. As Mendel [4] points out, "Engineers usually hate wasting time and effort. After all, time is money, and engineering is always under very tight deadlines".

The use of an outdated model or input data can result in incoherent results and the retardation of the engineering process. In the worst-case scenario, this could result in test failure or on-site forced outage. Our improved platform reduces the occurrence of these issues.

Workflow automation

The framework also enhances workflow automation. This was accomplished by providing a single tool to various engineering teams which allows for better technical support and synergy between users and results.

Automated workflows have several advantages. First, they are known to be faster. This is mandatory in order to perform more design iterations in the limited amount of time allotted by the market. As engineering time is valuable and expensive, it is in the interest of all to optimize productivity. As a consequence of automation, non-value-added tasks such as model gathering, data collecting, and run setting are greatly reduced. As such, the time saved can be reinvested in results analysis or technology development. Lastly, humans are more prone to errors than computers. Reducing non-decisional tasks for an operator also means reducing the risk of errors, often synonymous with wasting valuable time.

Furthermore, in-house framework is advantageous to the engineers using the resulting platform. *Ouellet & al.* [5] stated that “aerospace enterprises handle specific and complex design problems that require and rely on in-house specific extensive knowledge. No commercial CAE, CAD and FEA tool fully addresses all of these design problems”.

For this reason, all modules and submodules of this framework were developed, tested and validated with the expertise and requests of the engineers benefiting from this platform.

On-site Deployment

This framework concept has been deployed to the SIEMENS Aeroderivative Gas Turbine division. Single discipline activities have been defined and developed. An activity can be executed on its own or link to one another creating a workflow. Thanks to the PLM integration, multidisciplinary workflows can either be executed locally by a single user or collectively. The PLM database will ensure that all data are the latest relevant data applicable.

All activities are developed in an Object Oriented Programming (OOP) context to facilitate MDO integration. Iterative and optimization processes will be executed in separate modules of the framework.

This architecture is explained in a simplified scheme in *Fig. 1*.

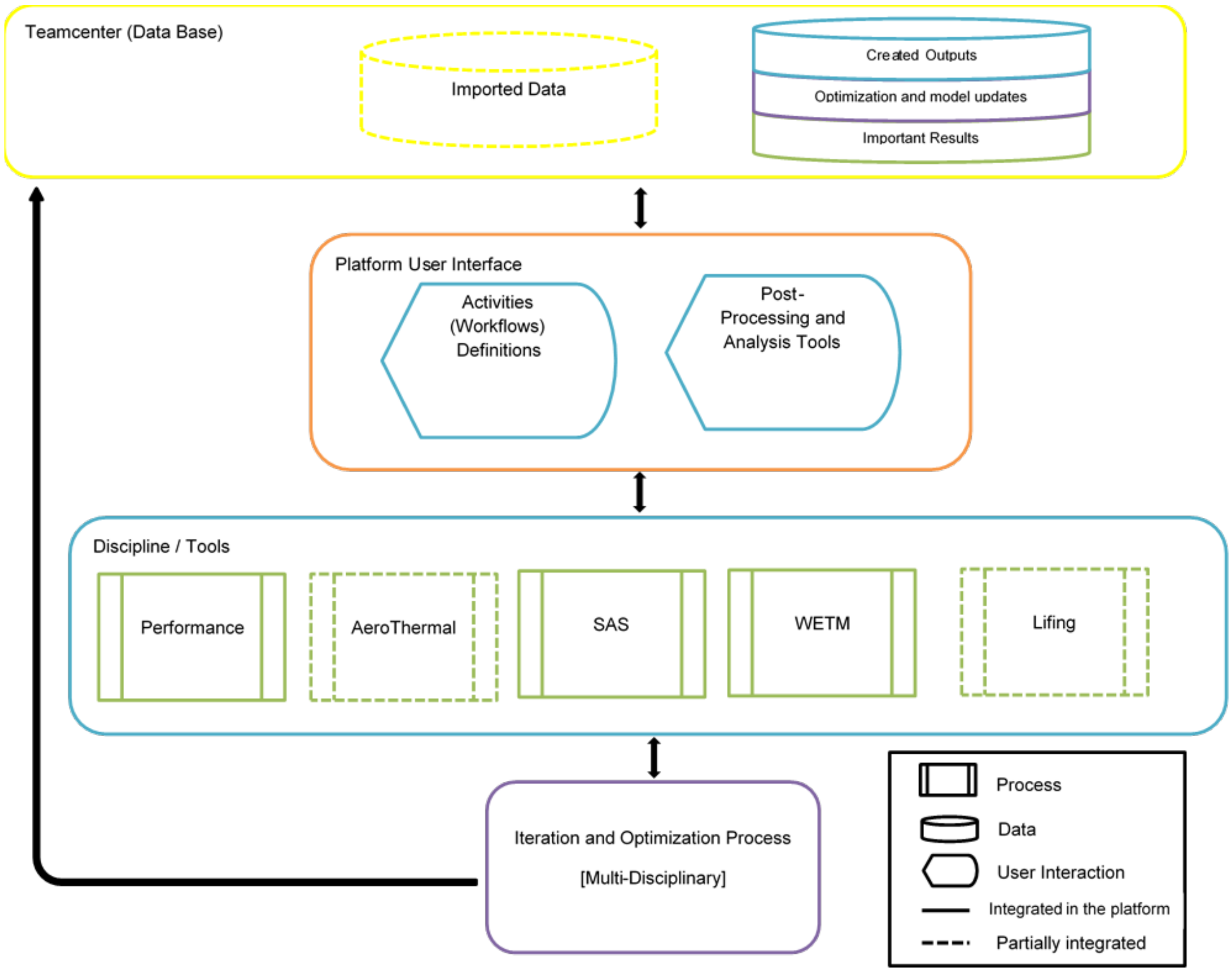

Figure 1 Design & Analysis platform concept applied to SIEMENS aeroderivative gas turbine development

*WETM: Whole Engine Thermo-Mechanics

3-Automation

Automating a workflow is not as trivial as it might seem. A fully automated workflow is impossible. Many parameters are highly sensitive and a slight difference can induce a large consequence. The user must be able to have access to these parameters in order to work efficiently. Unfortunately, allowing user setups reduces the automation, and when user inputs are required the process becomes longer. To address this problem, we conducted a careful analysis of the process in order to determine which tasks can be skipped, modified, or automated, and which will inevitably require user inputs. This precise analysis of the process is crucial for developing a properly automated workflow.

Workflow definition and analysis

Mapping the workflow was the first step toward improving the overall workflow. This has been done with an open-source software: *Cambridge Advanced Modeller (CAM).* Whereas the whole gas turbine design workflow has been mapped, the discipline of interest for this paper is the secondary air system. Therefore, only the results regarding this discipline will be explained. However, the reader should note that the methodology is the same for all disciplines.

Fluid systems analysis is required to assess whether the secondary air system will be able to provide the required amount of cooling air and sealing to the turbine components such as blades at a given operating point. In addition, SAS engineers are responsible for bearing load calculations to assess the integrity of bearings.

Project requirements determine the type of operating conditions to be analyzed, the model to be used, in addition to helping set up boundary conditions. The latter requires different type of data such as performance and aerothermal. Performance data is mainly composed of pressures, temperatures, flows across the main path stations, and shaft speeds of the gas turbine. This data is complemented by aerothermal data which sets SAS objectives and boundary conditions in terms of temperatures and flows for cooled components.

The workflow analysis was decomposed in two steps, a theoretical analysis and a practical analysis. The first focuses on software user guides, the company's best practices, tutorials, and defining a first draft of the workflow. This step has the advantage of gathering functionalities that the user might not be aware of and integrating them into future developments. The second is a practical analysis. This consists of sitting with users, conducting interviews, and shadowing them. The purpose of this observation is to describe a more realistic workflow, compile the flaws, and determine the automation suggestions of the users.

The workflow is separated into three sub-workflows: pre-processing (configuration of the model), process (execution of the model) and post-processing (analysis of the results). For each, a list of good candidates for automation has been established. Categories have been defined in order to determine the correct form of automation. The categories include:

Repetitive: Task that must be done multiple times during a single execution with no required engineering skills. Full automation is applicable in this situation.

Time Consuming: Long task that requires engineering inputs. Necessary, but should benefit from a better and more efficient UI to handle user inputs. To correct this issue a semi-automation is proposed.

Inefficient: Task requiring inputs that are already implicitly imported to the workflow. It often leads to errors through inconsistent results. To solve this issue an automation with metadata is proposed, using the inputs as a source of automation.

Unnecessary: Task often related to old best practices that do not bring any specific knowledge or value. Such tasks should be eliminated if possible.

This methodology has been adopted for the SAS workflow. Fig. 2 represents the results of the analysis and actions that have been taken.

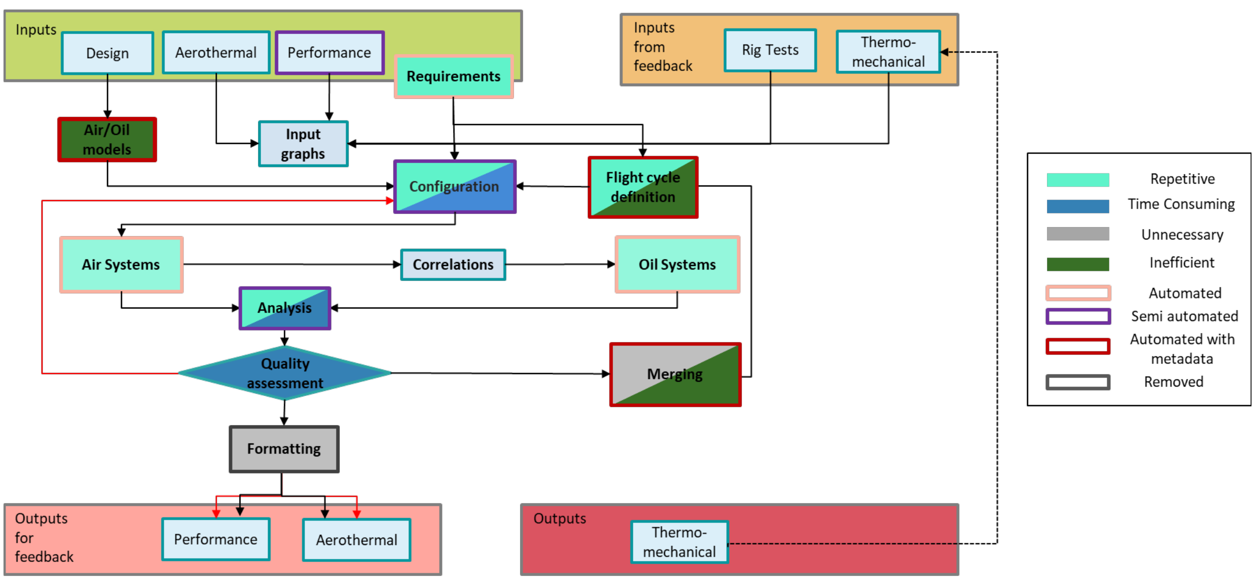

Figure 2 SAS workflow demonstrating the task-issues and their solutions.

Automation

<u>Pre-processing</u>

This step is the most important step for the SAS calculation process. It must be done carefully in order to obtain consistent results. This is where the success or failure of a run is determined. The engineer analyses the project requirements as well as task requirements, which are either pushed by performance willing to try a new engine configuration or pulled by other disciplines interacting as part of a feedback loop. The model configuration requires engineering judgement and should therefore accept user inputs; to accomplish this, a semi-automated process is proposed.

This semi-automated process was designed so that instead of manually parameterizing the execution, the user fills in a UI that will automatically configure once the execution is started. Thanks to performance metadata, the type of engine is passed to the configuration UI. The latest model in production is automatically selected as well as any complementary files. The same logic applies to the "flight" cycle data.

Processing

Once configuration is fed from pre-processing, the execution of the model starts. As this phase is very long and demanding in "clicks", it has simply been automated. The tool is run in batch and the execution of a full rating curve (cluster of operating points representing as much as possible the design envelope) is launched with one click at the end of the pre-processing. Calculating boundary conditions and solving the network is fully done in batch without user input. Once all results are created post-processing can begin.

Post-Processing

This phase is when the results are analyzed and conclusions can be drawn. Too often engineers suffer from the overwhelming amount [6] of data that is created and stored at various places for which formatting is necessary; this is why these tasks have been semi-automated. Consequently, a UI appears when the results are available so that the user can immediately identify failed runs in the status section. Furthermore, individual output files can be read directly from the UI without needing to open additional software or text editors.

The majority of analysis time for the workflow is spent on post-processing. However, it has been witnessed that a great proportion of that time was spent formatting data in order to plot the results. The proposed tool offers the possibility to plot all graphs from the main UI; both formatting and data gathering have been eliminated. The user can focus entirely on which parameters to plot.

Another important functionality that has been implicated is the use of presets. The user can save presets that will automatically display the desired graphs depending on the task being run. These presets can be saved globally and shared with other users.

Once the results are validated, the user can save them, and they will automatically be sent to the PLM (Teamcenter) to become accessible to other users that might need them for their tasks.

Automation with meta data

As mentioned above, it is highly inefficient to select data while these are implicitly already selected. Indeed, in the SAS context, the engine type is embedded in the performance data, therefore the user is spending unnecessary time gathering the right model and all other inputs from the requirements.

As a solution, metadata have been joined to the performance output files when they are created for automation. When the SAS tool is launched, it reads the metadata attached to its inputs and pre-selects the right model and other configuration parameters.

Metadata are stored in JavaScript Object Notation (JSON).

An example of the metadata structure is given below:

{ Model Version: <string>,

Engine Configuration: <object>,

Ambient Condition: <array>,

Task Description : <sting>

….}

Automation with metadata appeared to be very effective in the pre-process and post - process phases of the workflow, eliminating some non-value-added tasks and reducing errors.

Results

To quantify the benefits associated with this study, experimentations have been performed. As the tool is used by both expert and multidisciplinary engineers, subjects were chosen in both fields. A total of 8 engineers or engineering students participated.

They were asked to perform the above-mentioned SAS workflow using both methods, the original and newly proposed. This was done twice to reduce the confound of a learning curve since not all subjects were familiar with both methods. All runs were similar but with slight differences so that users had to modify inputs and be fully focused on their tasks. Some users had to run the workflow for 2 ambient conditions, others had only one whereas some tried both.

Engineering decision was not captured as engineering tasks were not automated and should not be reduced or fasten.

The results were intended to be interpreted as observations that could lead toward the acceptance of such a tool.

The first observation of interest was the time repartition for a power-curve calculation at one ambient condition. The overall time spent on the task is two times greater for the original method (Fig. 3). Furthermore, it was witnessed that the proposed method performs the best in the post-processing phase (around 2.5 faster for the new method). This suggests that more significant results will be observed as the tool has been developed to perform even better for a greater number of ambient conditions.

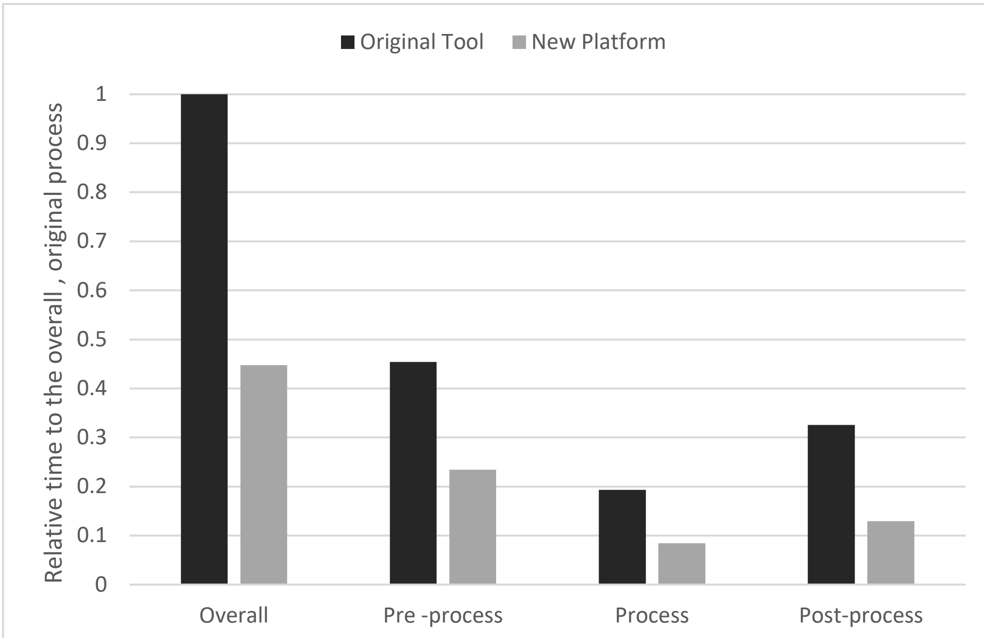

Figure 3 Relative time of processing using both the original tool and the new platform. Power curve calculation performed for one ambient condition.

The second observation of interest is the repartition of post-processing time. The advanced visualization tool has been developed to enhance plotting tasks and reduce errors. It has been observed that while the original method post-processing time was split equally between formatting the data and plotting the desired graphs, the proposed tool is only impacted by plotting graphs. As shown in Fig. 4, plotting graphs is now twice as fast as the original version. As data formatting has been removed thanks to automation, the overall activity is four times faster. This analysis only considers the graphing aspect of the post-processing and no output analysis.

Such a time reduction has allowed engineers to spend more time on the data analysis rather than wasting time gathering and formatting data. This directly serves the primary objective of this paper.

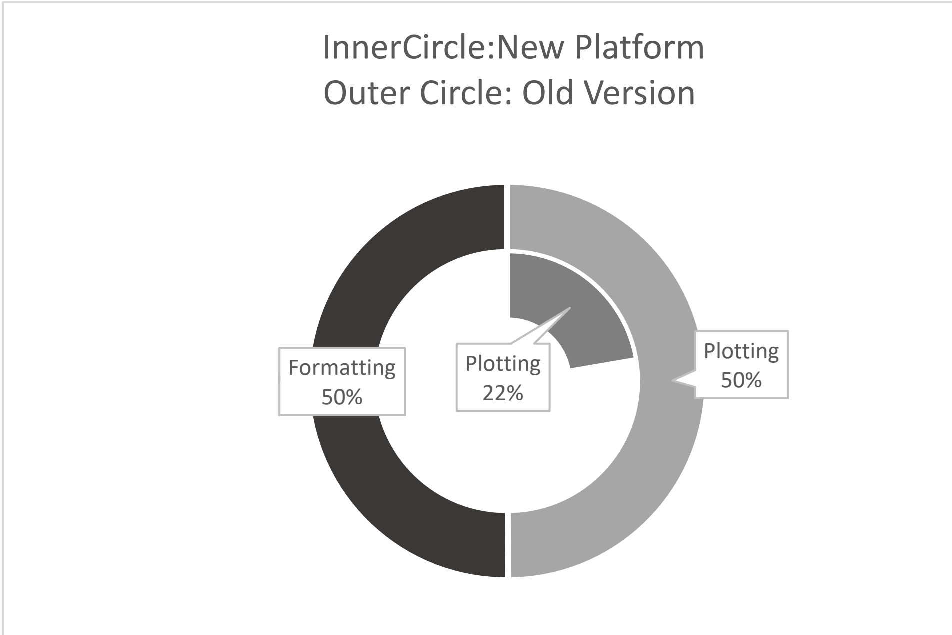

Figure 4 Comparison of the repartition of post-processing time for both methods

While previous results are based on calculations from one ambient condition, it is important to remember that gas turbines operate on a wide range of ambient conditions. The tool has been developed to enhance design space exploration. Multiple ambient conditions can be run at the same time. The overall process has been compared at one and two different ambient conditions. From those results a projection has been established. It is estimated that the new process would be 35 times faster than the original (Fig. 5) if 100 conditions were run. Theoretically, this number would be even greater as the original process requires user input at all times. Indeed, the proposed tool can be launched at any time and will run continuously until results become available. This performance will allow any team to perform calculations for more design points and enhance result accuracy.

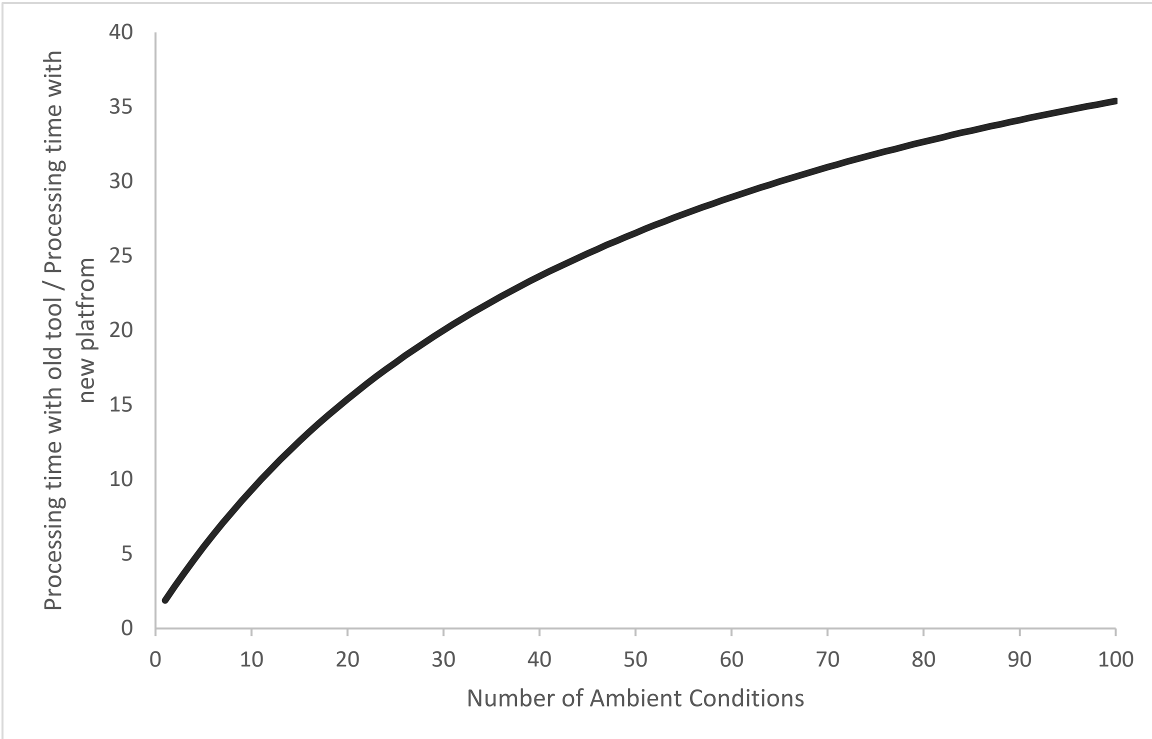

Figure 5 Projection of the performance tool with multiple ambient conditions

4-Feedback Loop

Model Convergence

As performance and SAS mutually use inputs and outputs from each other, some inconsistencies between them can appear. Every hardware or modelling modification should result in a full model convergence to ensure the most accurate results possible. This process being very demanding on engineering time, cannot be done sufficiently often to reach the full desired convergence.

The performance model, in order to calculate the data across the compressors and turbine, needs to estimate the amount of air that will be taken from the main path and brought to the SAS at a given station. These fractions of the main flow are called bleeds.

However, these bleeds are fully determined by the SAS model not the performance. Performance must therefore use estimations of the future bleeds. This can lead to inconsistencies as the SAS model centers its results on the performance data.

In order to ensure model alignments, an iterative process has been set up. SAS calculated bleeds are fed back to the performance model until both the performance initial guess bleeds and SAS bleeds converge (figure 6).

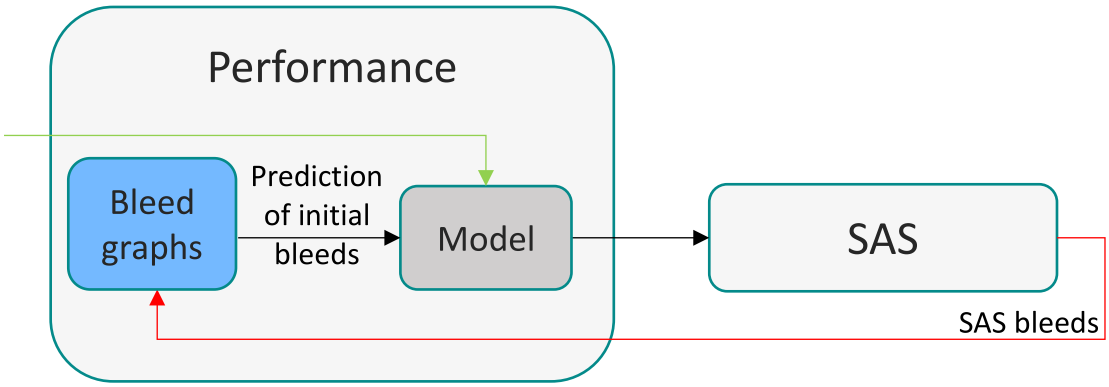

Figure 6 Performance and SAS model interactions

Moreover, until now, the convergence activity is performed at one ambient condition and at one power configuration. Calculated bleeds are then scaled to other points throughout the design envelop. This imply that the convergence would be the most accurate for a single design point.

Completing more convergence processes, to a tighter convergence criterion for more design points will enhance accuracy of the results. However, because of time and computation cost, this cannot happen without automation and a proper convergence strategy.

Automation Workflow

As the user should always be able to parametrize the model, both performance and SAS models should be run at the desired points before running the convergence tool. The generated bleeds are then automatically gathered from the SAS results formatted and processed to be fed back to the Performance Model. The iterative tool takes over the process until the convergence is reached. Finally, when convergence is reached, bleeds are set as new standards for the performance model.

The convergence module has been developed thanks to a Multidisciplinary Design Analysis (MDA) tool.

MDA approach

In this section, we study the multidisciplinary interaction between the engine performance and the secondary air systems. The interacting disciplines are coupled through the secondary air's bleed generation y and the performance data x, where the earlier feeds forward to the secondary air system, and the latter feeds back to the performance as depicted in Fig. 7. We solved this MDO problem using a distributed individual discipline feasible (IDF) formulation and non-hierarchical Analytical Target Cascading (NHATC) coordination [7]. Two subproblems are formulated, respectively responsible for the calculation of the coupling variables x and y. This implies that two copies of x, denoted x_1 and x_2, are created and introduced to subproblems 1 (performance) and 2 (SAS), respectively. A penalty function $\phi_x(x_1 - x_2)$ is then added to the objective function of each subproblem to ensure that $x_1 - x_2$ converges toward 0. Since x_1 is the result of the analysis of subproblem 1 x_2 is an optimization variable in subproblem 2. Similarly, for y, two copies are defined. A penalty function $\phi_y(y_1 - y_2)$ is added to the objective of both subproblems to ensure consistency. Since y_2 is the result of the analysis of subproblem 2, y is an optimization variable in subproblem 1. We use the alternating direction method of multipliers for the iterative solution (coordination) of the two subproblems as described in [7].

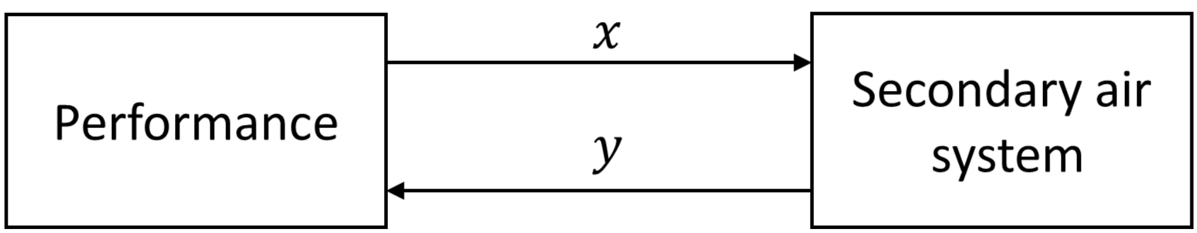

Figure 7 Bleed's generation distributed MDO problem

5-User Experience

The developed tool is a general tool meant for a wide diversity of engineers. It will have broad uses and will remain accessible and comprehensible to all. In order to perform the task it has been designed for, the tool must be adopted by users. However, this adoption does not only rely on pure result accuracy. Indeed, intangible effects such as user acceptance will determine a tool's success or failure. This intangible effect can be referred to as user experience (UX).

There is no doubt that researchers produce the most powerful, accurate and precise tools, but that does not suggest that these tools will be widely used. Indeed, too often because of their complexity they are difficult to manipulate and require expertise in the field. As a consequence, these tools fail to be widely used within the industry, as they are poorly accepted by users. Engineers and companies will often choose what they know (simplicity) rather than most what is most effective (advanced technologies).

Our proposed tool was developed with UX best practice concepts in mind at every stage of the process. This will help unite research and industry, with industry benefiting from state-of-the-art tools. Such development is greatly enhanced by on-site AGILE development. Developing the product with the support of users and early deployment assured a soft transition, acceptance, and a necessary product-market fit.

Moreover, the proposed platform is designed for multidisciplinary purposes. This implies that discipline specific experts will use it as well as multidisciplinary engineers who have general knowledge across different gas turbine design fields. It is a great challenge to have both profiles coexisting through the combination of expertise and general multidisciplinary executions.

Based on Nielsen's work [8], five keys concepts have been selected and have motivated UX development throughout the project: *User information*, *User freedom and Error management*, *Documented help* and finally *tool standardization across modules*.

User information

Automating the process will create longer wait times for the user. Since the pre-processing is done before the first execution, the tool will automatically run until results become available. This will generate a significant amount of time without user input. Therefore, the user can use their time more efficiently while they monitor the status of the execution.

To that end, a progress bar has been implemented. It displays the estimated remaining time and the task the tool is completing. Myers [9] demonstrated that users prefer progress indicators. Moreover, users are more willing to accept time variability when they are aware of what is taking place. This is of great interest since not all speed parameters are the user's responsibility (network speed or computer speed).

Another important aspect, is that in a multidisciplinary context, non-experts are also using the tool. Some of them might not be fully aware of the time every step requires and might easily think that the process has crashed. Alternatively, they might get bored waiting for the tool to produce data, while a long process is running, which is even worse [10].

Keeping the user informed of what is happening and when the process should end will also enhance productivity as the engineer can better optimize her time.

User Freedom

Too often design automation results in the reduction of the user's freedom as the tool can only perform specific tasks. This can greatly reduce user acceptance as they can feel "trapped". Therefore, the SAS module is designed to leave as much freedom as possible to the user while reducing possible errors. Instead of limiting the user's liberty, the tool focuses on possible errors and warns the users when their actions might lead to one.

Documented Help

The authors strongly believe that any user should be able to understand how the tool works without having to refer to any documentation. However, this is not realistic. Help menus are available for the users at different levels and will help them to easily find the answer they are looking for. Specifically, in addition to the help menu and tutorial section of the platform, every module also has one which is accessible from the module's UI. SAS' menu has been designed during development phases and thanks to users frequently asked questions. In order to fit the *Tool Standardization* section, all help menus are built around the same architecture.

Tool Standardization

During the workflow analysis the authors have witnessed that all tools are widely different and do not follow a strict pattern. For instance, a different logic is used to operate tool actions such as starting the execution, importing data, saving results or gathering output files. This can be confusing for new users that have to adapt to every tool and know their specificities. All this re-enforces the attachment of one user to a specific discipline mostly focused on one tool and as a consequence reduces multidisciplinary activities.

All modules respect a specific logic of how to start, save, import inputs, and plot graphs. The multidisciplinary advanced visualization tool allows the user to post-process the tools more easily with the same logic across the platform. Even though some specificities might apply to each tool, the core and architecture remains the same across the platform.

The proposed modules are developed respecting the same minimalist design pattern. No irrelevant information is displayed to help an effective analysis process and keep the users undistracted from unnecessary solicitations. However, every piece of information is still available on request to respect *User Freedom*.

7-Discussion

The above described tool has shown great potential, as it proved to be faster than the original workflow, it led to the design cycle being ultimately shortened, and allows for more design iterations happen. This inevitably enhances accuracy of the results and leads to improved products.

The tool could become even faster as parallel computing is under development. Post-processing could also be improved by allowing the user to start their analysis as the results are created and not at the very end.

Artificial intelligence could be useful to help the user comparing their results with previous results know to be correct by finding the best fit.

The MDO platform will gain efficiency as the number of integrated tools within the platform increases, the proposed MDO architecture would show its full potential by operating more than two disciplines at the same time.

References:

[1] Tarkian, M. (2012). Design Automation for Multidisciplinary Optimization : A High Level CAD Template Approach (PhD dissertation). Linköping. Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-81880

[2] Ebert,C (2013) Improving engineering efficiency with PLM/ALM, Software & Systems Modelling Vol2 Issue3 p 443.449

[3] Arun Ramamurthy et al (2017) DEVELOPMENT OF A SysML FRAMEWORK FOR GAS TURBINE DESIGNUNDER UNCERTAINTY, GPPS proceedings

[4] Mendel AF. (2011) Why Care About PLM?. ASME. Mechanical Engineering.;133(03):42-43. doi:10.1115/1.2011-MAR-5.

[5] Ouellet, Yannick & Savaria, Christian & Roy, François & Moustapha, Hany & Garnier, François. (2017). A Preliminary Design System for Turbine Discs. International Journal of Turbo & Jet-Engines. 10.1515/tjj-2016-0067.

[6] Mcmanus, H & Haggerty, Al & Murman, Earll. (2014). Lean Engineering: Doing the Right Thing Right.

[7] Tosserams, S. & Kokkolaras, M. & Etman, L.F.P. & Rooda, J.E. (2010). A Non-hierarchical Formulation of Analytical Target Cascading. ASME Journal of Mechanical Design, 132(5).

[8] Nielsen, J. (1995). 10 usability heuristics for user interface design. Nielsen Norman Group

[9] Myers, Brad. (1985). The importance of percent-done progress indicators for computer-human interfaces. ACM SIGCHI Bulletin. 16. 11-17. 10.1145/317456.317459.

[10] Foley, J.D., & Wallace, V.L. (1974). The art of natural graphic man-machine conversation. *COMG*.