Proceedings of ASME Turbo Expo 2011
GT2011
June 6-10, 2011, Vancouver, British Columbia, Canada

GT2011-46641

EXTENDING THE FLEXIBILITY OF OBJECT ORIENTED SYSTEMS MODELING ARCHITECTURES

Steven Sirica
Pratt & Whitney
400 Main Street
East Hartford, CT 06108
USA

Igor Fuksman
Pratt & Whitney
400 Main Street
East Hartford, CT 06108
USA

ABSTRACT

When adopting an industry-wide generalized modular based system simulation tool (such as the Numerical Propulsion System Simulation – NPSS™), the generic modules provided with the system do not typically provide the fidelity required for detailed engineering design and analysis nor do they necessarily align with company specific methods. The challenge is to develop company specific replacement modules that would promote standardized methodologies while still providing the flexibility required for unique methods and / or joint venture teaming arrangements. Pratt & Whitney (P&W) has developed such a system within the NPSS™ framework that leverages the intrinsic Object Oriented (OO) capabilities of this tool without compromising the integrity of the standard methodologies. This flexibility easily allows a single system to be used for engine systems modeling from Concept Initiation through Fielded Product Support. Additionally, the structure and capabilities of this newly developed system provides the means for a significant reduction in the manpower required to maintain / upgrade this toolset.

INTRODUCTION

The Numerical Propulsion System Simulation (NPSS™) was developed during the end of the twentieth century by NASA and industry partnership [1]. It has since evolved to an industry based consortium [2] whose structure is shown in Figure 1.

Note that the NPSS™ modeling environment is used extensively throughout this document. NPSS™ was chosen because it is an available product and is in widespread use at P&W. The methods developed within this system and described within this paper are directly applicable to other object oriented systems.

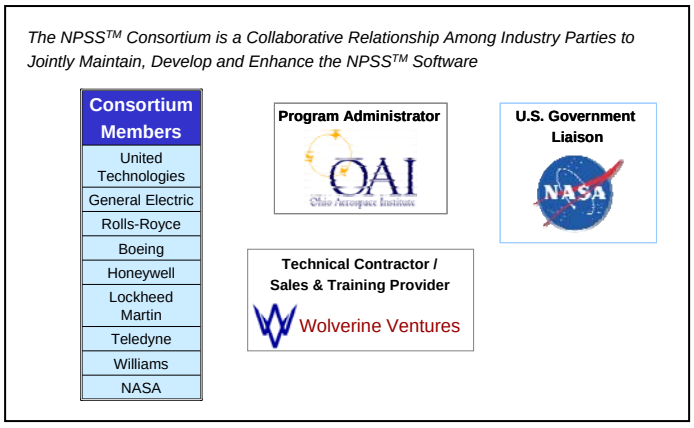

Figure 1: NPSS™ Consortium Composition and Structure

From the beginning of the development effort, one of the major tenets was to build a modular system that would enable the systems engineer to easily create simulations of complex gas turbine engines to predict and analyze the aero-thermodynamic engine performance. These simulations are generally configured around what is commonly referred to as a 0-D level where each major component is represented with the control volume surrounding the entrance and exit of the respective component. For instance, a High Pressure Compressor (HPC) would be represented as a single compression process regardless of whether it was a 1-stage compressor or a 12-stage compressor. In order to accurately capture the performance of a component while representing it as a simple compression process across the control volume,

another major requirement of this modular system was to provide a means to easily connect detailed analysis adjustments to each component in order to better represent the physics of the system. The inherent capability to easily integrate all of these components and their sub-systems results in a flexible and highly capable system analysis aero-thermodynamic engine simulation.

NPSS™ OVERVIEW

The basic NPSS™ system contains a suite of component modules (referred to as Elements) that can be configured to simulate the performance of the majority of air-breathing Brayton Cycle gas turbine systems [3]. These Elements consist of the major flowpath components (Inlets, Compressors, Burners, Turbines, Nozzles, Shafts, Bleeds, Splitters, and Mixers) as well as a primary means of connecting these components via intrinsic Fluid and Mechanical Ports. Figure 2 depicts schematically how a simple Turbojet would be represented in this system.

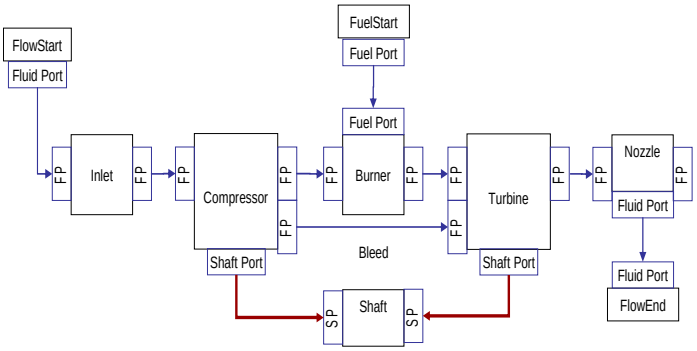

Figure 2: Simple TurboJet Schematic Modeled in NPSS™

Each of these Elements contains the necessary calculations to reflect the physics across the component, but in true Object Oriented Programming (OOP) fashion the calculations are part of the Element (object) but the data required for the calculations are external to the Element. An example of this is a Compressor Element that contains the equations to calculate its exit temperature based on the component pressure ratio and efficiency, but the calculations of pressure ratio and efficiency using maps and scaling are provided external to the Element.

In an effort to keep the components modular and keep the requirements for source code maintenance to a minimum, the system was architected with what are called Sub-Elements. Sub-Elements connect to and communicate with Elements by means of intrinsic Sockets. Only one Sub-Element can be connected to any given Socket. Sub-Elements can have other Sub-Elements connected to them, again through the use of Sockets.

An example of the use of an Element with a Sub-Element, shown in Figure 3, would be taking the previously mentioned High Pressure Compressor Instance of a Compressor Element that contains the equations required to calculate the exit temperature based on the component pressure ratio and efficiency. Instead of embedding the table read of the component map and any associated scaling directly within the Compressor Element, a Map Sub-Element was created that performs the necessary calculations and passes this information to the Compressor Element. The benefit of this approach is that if there were a need to use a different Map methodology one would simply connect the new Map Sub-Element into the Compressor Element through the Socket.

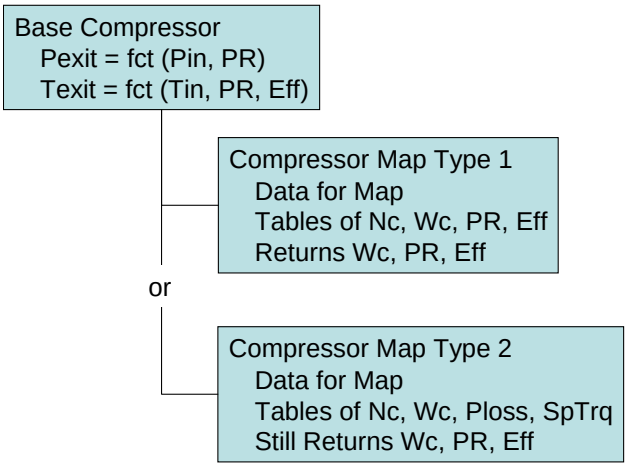

Figure 3: Example of Separation between Elements and Sub-Elements

DEFINITION OF NEEDS FOR ENHANCEMENT

The Elements and Sub-Elements provided with the baseline system are well developed for the basic representation of the respective components. However, it has been found by some companies that these Elements and Sub-Elements do not adequately match the requirements or meet a particular company's needs. This can be due to the fact that a) the supplied method does not fit in with a company's legacy toolset or b) the level of fidelity does not match a particular company's validated methods.

ISSUE 1 – INSUFFICIENT FIDELITY

The potential shortcomings of the supplied Elements and Sub-Elements created the need for companies to develop their own proprietary versions. Typically when a company created a customized version of an element, they would keep the same base name and add a suffix that reflected the lineage of the element. For example, the base compressor Element provided with the system is called Compressor but one customized by P&W would be called CompressorPW. This syntax also applied to all customized Sub-Elements as well.

The framework of connecting Elements and Sub-Elements provided a clear way of communication and allowing for separation of the various processes and methods. These customized Elements followed the same style of the base

Elements and used single Sockets for communication between Elements and Sub-Elements, which is shown in Figure 4.

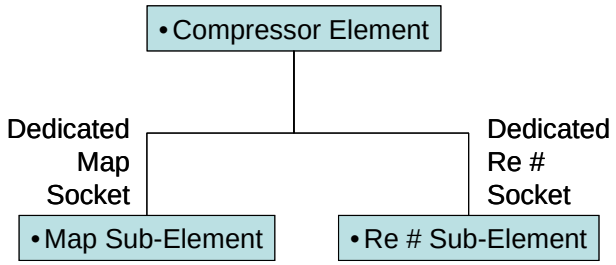

Figure 4: Example of Dedicated Sockets for Communication

During the early phases of adoption and migration from legacy modeling systems to an object oriented system, this approach was very manageable. However, one point that was becoming evident was that many Sockets were required for a main Element to accommodate all of the sub-system effects. Below is a list of the Sockets provided with the baseline Compressor Element [2] compared to that of a custom CompressorPW Element.

Baseline Compressor Element Socket List

- S_map – Compressor Map
  - S_eff – Base Map Read (eff based)
  - S_hum – humidity effects
  - S_Re – Reynolds Effects
- S_Qhx – Compressor Heat Transfer Effects

Custom CompressorPW Element Socket List

- S_map – Base Map Read (multiple formats)
- S_Re – Reynolds Effects
- S_wet – Humidity Adjustments
- S_cal – Calibration Adjustments
- S_adj – Adjustment Effects
  - S_bleed – due to bleeds
  - S_clear – due to clearances
  - S_deter – due to deterioration
  - S_VSV – due to off-nominal Stator Vane
  - S_untwst – due to Untwist
  - S_prof – due to profile shifts
- S_SM – Surge Margin Calculations
- S_warp – Warpage Effects
- S_Qcase – Case Heat Transfer Effects
- S_Qblade – Blade Heat Transfer Effects

As can be seen from this comparison, the complexity of the Element and Sub-System Effects handled through the Sub-Elements is much larger than the original design of the module.

The initial migration to the customized object oriented system aligned well with validated legacy methods and was an adequate replacement for a period of time. However, as the engineering community became more comfortable with this toolset and their understanding and learning of the system grew, there was a desire to leverage this flexible system to add fidelity to the simulations. This requirement for increased fidelity was being driven by a requirement to improve predictive capability and reduce engine test time.

To fill this need for increased fidelity, the user community was starting to get creative in their use of the existing methods. The use of doubling and tripling the effects into a complex table or converting the tables into complex functions where multiple effects were modeled was becoming widespread. This approach worked but there was loss in visibility of the effects since the individual effects were rolled into single value adjustments. Additionally, the system became quite challenging to make changes to these composite tables / functions.

As an example of the creativeness of the user community, imagine a basic Nozzle Element that calculates the flow through an orifice for a given area and nozzle pressure ratio. One of the standard adjustments to a Nozzle is the Discharge Coefficient (Cd), which is a modifier on the nozzle area. The Cd is calculated in a Sub-Element and passed through a dedicated Cd Socket (S_Cd) back to the Nozzle Element. The standard format of the Cd implementation is a simple one dimensional X-Y table that has nozzle pressure ratio on the X-Axis and Cd on the Y-Axis. This works fine in the early prediction phases of an engine program. However, once an engine has been tested, there is often a need to update the Cd characteristic to calibrate the simulation to measured test results.

In the case where the adjustments to the Cd correlate to the same independent as the base table (in this case, nozzle pressure ratio), the creation of a composite table external to NPSS is fairly straight forward and the updated table can be placed into the simulation. Note that in this approach the visibility to the calibration effects are buried into the new table and are not available to the using community.

Another more likely scenario is that the Cd calibration is not a function of the same independent as in the base table (nozzle pressure ratio) but is a function of another nozzle parameter. Let's say that the Cd calibration adjustment to the base correlates as a function of the inlet corrected flow of the nozzle. In this situation, the user could either **a)** create a more complex two dimensional table that is a function of nozzle pressure ratio and is layered as a function of inlet corrected flow, or **b)** convert the basic Cd table into what is referred to as a Function (intrinsic NPSS™ capability) and inside of this function write customized methods that read the multiple tables and calculate the combined effect. Similar to the situation where a composite table is created external to the basic element, visibility into the calibration effects are lost to the using community. Additionally, since non-standard methods are being used, the setup time and potential for errors is significantly increased.

It was critical in the development of the custom Elements that they still communicate to the base system in the same manner as the baseline Elements and Sub-Elements. This allowed for a continued Plug-and-Play approach combined with mixing and matching of Elements (note that Sub-Elements are generally linked with a particular Element design and are not mixed and matched). This was particularly useful in joint ventures / partnerships. For example, imagine a teaming arrangement between Partner A and Partner B. If Partner A is responsible for the Low Spool design and Partner B is responsible for the High Spool design, each company could simulate their respective components with specific methodology of their company, but at the same time the components can still interact and communicate with one another in a seamless manner.

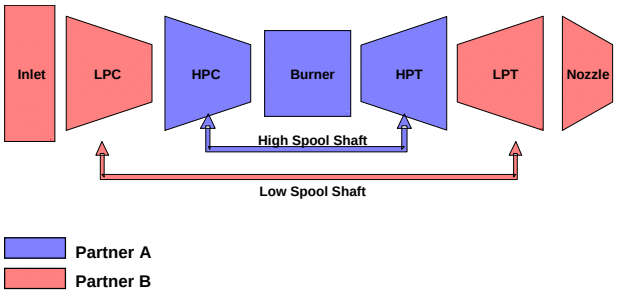

Figure 5: Example of Potential Joint Venture Model Split

ISSUE 2 – INCOMPATABILE "ADJUSTMENTS"

The NPSS™ system and its Object Oriented Framework provided a cross-divisional and cross-company flexibility that was not previously available in the air breathing gas turbine industry. The capability of this system enabled joint venture simulations with minimal effort. It also provided the means for Tier 2 suppliers (sub-contractors) and sister divisions within a company to supply drop-in modules of their components ready for direct integration within a system simulation. The one caveat to this approach is that unless a supplier or partner had the full system available and they were capable of developing their own Elements and supporting Sub-Elements, their effects had to adhere to the available methods.

A good example of this is the application of Reynolds effects to a High Pressure Compressor. In industry, there are several well established methods of applying Reynolds effects. One method is to adjust the compressor efficiency at constant compressor pressure ratio (note that this implies a shift in the available work of the compressor). Another common method is to adjust the compressor efficiency at constant work level of the compressor (conversely this implies a pressure ratio adjustment to the compressor). The standard method available in the customized PW Sub-Element applies the modifier to the compressor efficiency (as a function of the Reynolds Number) at constant compressor pressure ratio. If a partner's standard practice was to apply the Reynolds effect at constant work, they either had to create their own custom Elements or convert their effects into an equivalent approach that was compatible with the P&W custom Element.

NOVEL APPROACH

At this point, a decision had to be made as to whether to continue down this path or to re-architect at the Element level the fundamental concept of how effects are handled.

After examination of the shortcomings of the current approach, it became evident that an alternative solution was necessary. One key of this solution would be to leverage the architecture of the object oriented system to allow for the connectivity of an unlimited number of Sub-Elements through a single connection. Another key would be to devise a method that would allow mixing and matching of different effects in a layered approach.

UNLIMITED CONNECTIONS

In order to handle the request for an unlimited number of connections of Sub-Elements to the upper level Sub-Element or Element, it was necessary to devise a list based interface that would dynamically create and process the connections. This list based interface would have to loop through all of the created Sub-Elements. It would also have to be able to react as new Sub-Elements were created on-the-fly as part of the run-stream execution.

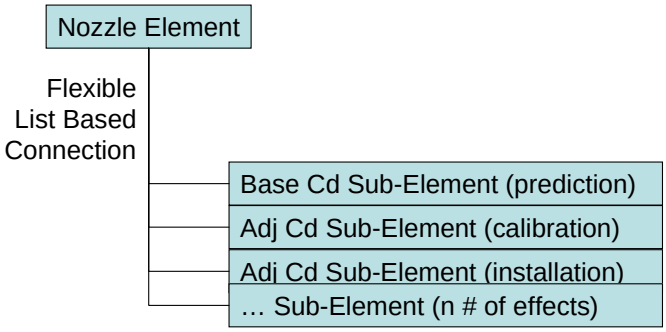

Figure 6: Example of Multiple Sub-Elements Linked Through a Single Connection

Addressing the addition of newly created Sub-Elements was straight-forward as the base system provides intrinsic features that aid in the discovery of such items. Dynamic deletion of Sub-Elements proved to be more of a challenge. Capturing the addition or deletion of items to the variable list interface was not an issue. The issue was in the actual deletion of Sub-Elements once created within the system architecture. This would often work well but configuring the simulation to do this requires extensive knowledge of the system. One runs the risk of there being the deletion of Sub-Elements without proper clean up upon their destruction. The Sub-Element would get properly destroyed but there may be instances where the Sub-Element's variables are referenced elsewhere in the model and that there is not easy way to identify and clean up

such references. A work-around solution was developed that allowed for the switching ON and OFF of a Sub-Element's behavior once it had been created, in effect making it transparent.

MIX AND MATCH OF EFFECTS

It was evident that a means of layering the various component effects was required to better capture and promote the use of a single modeling system yet minimize the proliferation of custom Elements and Sub-Elements. Once the decision was made to pursue a layered approach to the application of effects, a literature search [4, 5] was conducted by P&W to capture the most widely used methods in Industry. The idea behind this was to capture the majority of known methods once and for all and include them as options within the respective Sub-Elements.

Once these methods were integrated into the newly created custom Sub-Elements, they could be applied in any combination due to the layering aspect, a true mix-and-match application of effects. Coupling this layering approach with the revised Unlimited Connections method created a near infinitely variable system that could emulate all known modeling approaches available. Figure 7 shows an example of the application of the layered approach as applied to turbomachinery (compressors or turbines).

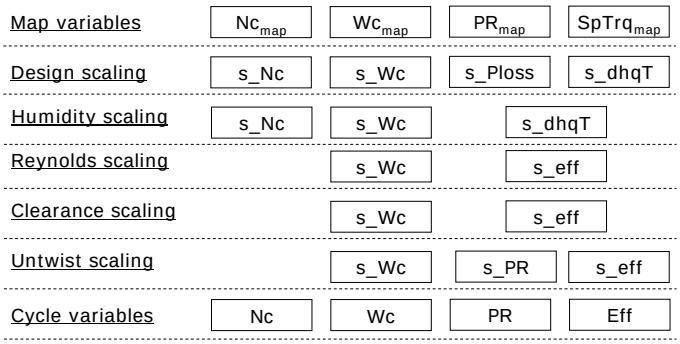

Figure 7: Layering Enables Mix and Match Approach

BENEFITS

This newly developed capability has enabled substantial time and cost savings associated with engine program development. Additionally, by providing a standard means of interfacing with component characteristics, it has significantly reduced the potential for errors in implementation.

In a recent joint venture, the P&W enhancements to NPSS™ capabilities have allowed for both P&W and the partner's components to be accurately modeled utilizing each companies preferred methodology. This is solely due to the fact that it is now possible to emulate the majority of known modeling methods within the P&W system. In this particular instance, the partner did not have any of their own custom Elements available. This new system effectively eliminated the cost and time associated with creating custom Elements and Sub-Elements. It is no longer a requirement that a partner develop expertise in the customization of NPSS™ just to leverage the benefits associated with it.

In another recent development program, the P&W enhancements were able to easily emulate a module supplier's method to reflect the component level aero-thermo performance. In this particular instance, the supplier had a requirement to adjust the performance of the turbomachinery to account for the impact of component clearances. The supplier's method was based on an approach that required multiple adjustments to the overall component performance (flow capacity and efficiency) and was a function of the individual stage characteristics. It should also be noted that the efficiency adjustments were required to be applied at constant work of the component, which is different than the default P&W method of delta efficiency at constant pressure ratio. This was easily represented using the unlimited connections capability and individual cumulative adjustments were created to match the supplier's methodology. Additionally, this also took advantage of the layering capability since many of the other effects in this module were done at constant component pressure ratio. Once again, this new system effectively eliminated the cost and time associated with creating custom Elements and Sub-Elements.

This highly flexible, easily customizable, modeling architecture will serve the needs of the system analysis community for many years and virtually eliminate the need for methods development and support. It also provides nimbleness to the systems engineer all the way from concept initiation through revenue service and field support.

REFERENCES

1. Lytle, J.K., "The Numerical Propulsion System Simulation: An Overview", NASA/TM-2000-209915, 2000.

2. NPSS™ Consortium Website http://www.npssconsortium.org/

3. NPSS™ Consortium, "NPSS™ User Guide", Software Release: NPSS_2.2, February 2009.

4. NATO RTO Technical Report 44 "Performance Prediction and Simulation of Gas Turbine Engine Operation", RTO-TR-044 AC/323 (AVT-018) TP/29, April 2002.

5. Kurzke, J., "Design and Off-Design Performance of Gas Turbines User Manual", Software Release: GasTurb 11, 2007.