Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 44 (2015) 125 – 134
Procedia Computer Science

2015 Conference on Systems Engineering Research

Georgia Aquarium Design Space Analysis and Optimization

Jeremy Coffeen, Frederic Jacquelin, Richard Kepple, Rock Mendenhall*, Michael Rodgers, Nicholas Roth, Richard Thompson, Richard Wise, Roger Wong

Georgia Institute of Technology, Atlanta, GA 30303, USA

Abstract

The Ocean Voyager exhibit residing at the Georgia Aquarium Inc. (GAI) is one of the largest reef gallon aquariums in the world, with a capacity greater than 6.2M gallons. Reef aquariums are closed systems and must compensate by 'turning over' their complete volume of water many times a day through biological, chemical, and mechanical filtration. Due to the Georgia Aquarium being a non-profit organization, GAI sought to investigate ways to maximize efficiency and lower operating costs. This paper will focus on using low-cost software solutions to perform trade space analyses and optimization directed towards the Ocean Voyager exhibit and related GA Aquarium life support and energy systems.

The software solution herein demonstrates a top-down System of Systems (SoS) to subsystem modeling approach that provides decision makers with interdisciplinary dashboard-level tools to visualize system design. The goal of the analysis is to provide executive level decision-making support for designing or enhancing existing complex systems and SoS. The analysis was performed as a capstone project by Georgia Tech graduate students progressing from cradle to finish in just 9 weeks to show the benefits of systems engineering to Georgia Aquarium staff. Integrating software SE tools into a single, aggregate model enables project engineers and decision makers to direct design directions with confidence.

Peer-review under responsibility of the Stevens Institute of Technology.

Keywords:

1. Introduction

Georgia Aquarium Incorporated (GAI) is a public aquarium in downtown Atlanta. GAI operates as a 501(c)3 non-profit organization. GAI relies on corporate sponsors, individual donors, volunteers, and public ticket sales to sustain operations. "The mission of the GAI is to be an entertaining, educational and scientific institution featuring exhibitions and programs of the highest standards, offering engaging and entertaining visitors' experiences, and promoting the conservation of aquatic biodiversity throughout the world" [1]. In executing this mission, GAI also has to be a good

Peer-review under responsibility of the Stevens Institute of Technology.
doi:10.1016/j.procs.2015.03.001

steward of their community contributions and thus desires to investigate ways to both improve efficiency and incorporate green energy solutions.

The Aquarium itself is a System of Systems (SoS) constructed of more than sixty interconnected fresh and salt water exhibits funded by multiple sponsors. The Aquarium's purpose was established at the onset, but the requirements, exhibits, energy technology, and hydraulic technology have changed over time. When GAI opened in 2005, it was unprecedented in that it was the largest Aquarium in the world. Perhaps due to its unprecedented water volume, when it opened to the public there were limited specific energy consumption goals and there were certainly no considerations for the use of green energy. Since its inaugural opening, the Aquarium has added exhibits, increasing energy demand and the energy cost keeps increasing. Newer aquariums of similar size are more efficient and have moved ahead with a host of green energy initiatives. The goals for GAI are not only to reduce their energy footprint and save money but also to become a leader in the use and education of green energy.

The GAI displays characteristics of an "acknowledged" SoS [7]. GAI has recognized goals, a Board of Advisors, and resources for the SoS, however, the constituent systems retain their own independent ownership, objectives, funding, and sustainment approaches [2]. The Georgia Aquarium architecture is diverse due to the number of sponsors, the number of exhibits, the unique requirements of each exhibit, and the complexity of sustaining marine wildlife. The aquarium architecture is organized around 7 themes: Ocean Voyager, Tropical Diver, and so on. However all exhibits share some common attributes: each of these themes includes a large-tank exhibit along with any number of smaller tanks and interactive displays. The general architecture for large-tank exhibits is shown in Figure 1. In this figure, each large-tank exhibit consist of actors (e.g., animal(s), corporate sponsor, researchers, and so on), an energy system, a water source, and a wastewater plant. Only the Ocean Voyager is shown here because it is the principal system of interest in this study. The other 6 themed exhibits have a similar block diagram description.

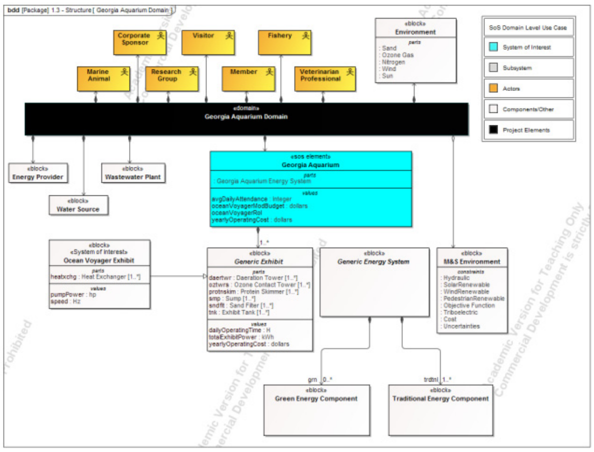

Figure 1. Georgia Aquarium Domain Level Block Definition Diagram (SysML)

2. Problem & Solution Strategy

Problem:

Using real world data, how can the Georgia Institute of Technology (GIT) Professional Masters in Applied Systems Engineering (PMASE) Team AquaTech, recommend changes to increase GAI's energy efficiency by 5% (threshold); 10% (objective) and lower operating costs? A secondary goal is to learn what the viable "green" alternatives and respective implementation costs for future upgrades. High-level constraints and project details are listed in the project overview, Table 1.

Table 1. Project Overview

Constraints:	1) Budget for proposed solution(s): $200-$400K
	2) ROI < 2.5 years
Timeline:	May 1 – Aug 1, 2014
What:	GIT PMASE Capstone Project
Members:	GIT PMASE Team AquaTech

The nature of the problem involves the aggregation of large multi-system design spaces and different engineering and cost simulation analyses into one coherent and synergetic decision making support process. Obtaining this solution calls for architecting a decision support process which would enable informed decision making across all stakeholders. The decision support process in making a system more efficient within technical and financial constraints can be complex and ideally would require the following:

- **Accessible:** easy to use processes allowing non-technical stakeholders to be included
- **Transparent:** trade off analysis is repeatable and unambiguous.
- **Traceable:** all requirements with an impact on the decision making attributes are understood by the decision makers, allowing them to adjust targets and constraints appropriately
- **Real Time:** promote "what if" scenarios and "decisions in the board room"
- **Robust:** in that uncertainties can be represented and their impact accounted for
- **Inclusive:** in that design variables and measures of performance are linked across each system
- **Accurate:** quantitative data and simulation driven, qualitative only when necessary (soft factors, etc.)
- **Cost Effective:** use open source languages and technical libraries

Various system engineering and design optimization approaches allow supporting such qualities and the critical issue is to successfully coordinate these approaches in a synergistic manner. While each approach has its unique pros and cons as shown in Table 2, the authors suggest an implementation of a "best of many" systems engineering disciplines using a real world problem. A roadmap is provided in Figure 3 for future improvements of these processes so they can be applied across a wide range of problems of similar complexity.

The executive level decision support tool often needs to be visual and simple to understand by abstracting much of the lower level modeling and simulation details away. The use of sampling, applying constraints and visualizing the design space is therefore a critical factor in allowing decision makers to be actively involved in the analysis, promoting consensus via common understanding of the process and the system behavior under study. Concurrently, detailed optimization methods can be employed to find a solution in a faster timeframe and handle highly complex behaviors existing between subsystems more systematically than a manual approach; especially when uncertainties are added into the mix and the number of variables becomes prohibitive for easy synthesis and visualization.

While numerical optimization may find the ideal solution faster – some techniques like heuristics may get stuck in local optima creating unease in letting an algorithm handle costly investment decisions without further checks while others, such as Exact methods will require long run time which is incompatible with our real time decision making support objectives. Stakeholders and decision makers prefer to be involved in, or at least have visibility in the analyses and decisions beyond simply providing requirements definitions. Asking them to implement the solution identified by a "black-box" optimization algorithm is asking a lot. Additionally, the identification of unrealistic and competing requirements is achieved and hence readily mitigated early on by the stakeholders. Complex Algorithms also require an expert user, costly solvers, to ensure that global convergence and repeatability in the proposed solution are achieved while at the same time maintaining the ability to examine the solution's sensitivity across uncertainties in a large design space. The process combines these approaches in a manner that helps detect inconsistencies and double checks solutions:

- QFD for example enables expert knowledge to provide high level directions for the design, hence providing for an initial simplification of the problem (reducing number of design variables for example) for the user so as to mitigate the overwhelming complexity of a problem (also applies to computing resources).

- Greedy algorithms allow Stakeholders to develop understanding over the design variables and their effects on the design space. Using a user driven iterative process around sampling the design space, setting constraints (top down and bottom up) and filtering for feasible designs down to a single or set of design solutions, the user maps out a decision process that is visual, reproducible and explainable. Our implementation combines different Global (Exact) Optimization techniques such as adaptive search and exploratory methods in a way that allow the user to be part of the process, hence deviating from Exact methods fully automated nature.
- Direct numerical approaches allow for the automation in finding a unique solution. Depending on the design space characteristics (linear, non-linear, discontinuous, etc...) these techniques may provide accurate results rapidly. However they may also provide a local optimum which can mislead the decision maker in the case of Heuristics techniques.
- Therefore, in mimicking exact methods processes in adding adaptive and exploratory phases to the optimization process, our proposed process can alleviate such problems. For example, the exploratory nature of the process can handle discontinuous space more effectively. However, the "black box" nature of this process may still not provide the full confidence in the solution that one may require, which led the team to choose for a path of collaboration between the techniques as opposed to choosing either extremes. Additionally, the algorithms' achieved speed also enables other types of analysis to be included, such as sensitivity analysis to test for robustness which is critical to decision making.

Table 2. Methods Comparisons Matrix

Methods	Example	Accessible	Transparent	Traceable	Real Time	Robust	Inclusive	Accurate	Cost Effective
Greedy Algorythm (Data Farming and Filtering or "brute force")	Visualization of design space (point clouds, mulit-sctatter matrix, filtering space with cosntraints)	+	++	++	+	o	o	o	++
AoA	QFD	++	o	o	++	-	+	- -	++
Direct Numerical Techniques	Method of Feasible Direction (MMFD), Sequential Programming (NLPQL), Mixed Integer Sequential Quadratic Programming (MISQP)	o	--	-	o	+	o	++	o
Global Optimization	Complete, Adapative Search, Branch and Bound	- -	-	-	-	++	++	++	o

Legend	--	-	o	+	++
	Reduced		Neutral	Improved	

Defining the Aquarium SoS architecture and subsequently establishing a SoS solution architecture is necessary because it decomposes the Aquarium's systems into individual modules which ultimately help manage the complexity due to the number of financial sponsors and the number of exhibits (Figure 2). The SoS Architecture also fosters the development of models and simulations as new technologies, modules, and interfaces emerge.

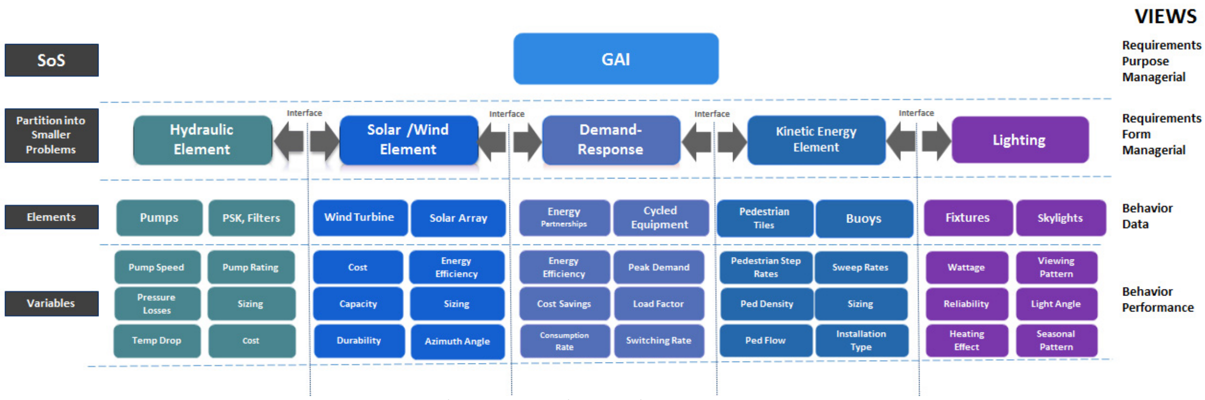

Figure 2. Georgia Aquarium Energy SoS

This decomposition is not easy because the Aquarium sometimes presents a radically different view to each stakeholder and these views must be integrated by a large number of highly-specialized disciplines. A list of the principal disciplines is provided below:

- Mechanical Engineering
- Electrical Engineering
- Software Engineering
- Industrial Engineering Systems
- Information Technology
- Marine Wildlife Conservation
- Cost & Financial Modeling
- Human Factors / Cognitive Systems

For analyzing the project problem and determining focus areas for optimization, PMASE Team AquaTech chose to first establish a SoS architecture. This resulted in an improved allocation of team-member resources by forming tiger teams to address the potential efficiency improvements that could be obtained from the following elements:

- **Solar Element:** Investigates solar mounted roof arrays, solar covered parking, solar roads
- **Wind Element:** Investigates roof mounted variable direction mini-wind turbines
- **Hydraulic Element:** Scoped to Ocean Voyager exhibit focusing on pump design, components flow losses and overall hydraulic infrastructure
- **Pedestrian Kinetic Energy Panel Element:** Investigates installing Kinetic Energy Panels (such as those produced by the British company, Pavegen) in specific areas of the aquarium to create electrical energy from visitors walking on the panels.
- **Lighting Element:** Investigates use of lighting technologies including 85W and 120W LEDs and the use of sensors to automatically control lighting in low traffic areas
- **Demand Response:** Partnering with energy providers like Georgia Power, a Southern Company, to reduce electrical consumption during peak demand.

3. Heuristics

Heuristics are simple phrases that are learned by experience and passed down to others to help guide the SE process. Design, budgeting, and other factors that influence design processes, are not always intuitive. Sticking to key heuristics was vital to help Team AquaTech stay on track within a compressed timeline for deliverables. Derived heuristics from external sources that can be applied to the GAI project are as follows:

- <u>Heuristic 1:</u> *"Systems engineering involves communications, critical to international partnerships, so before worrying about technical interfaces, make sure the integrated product teams and communications bandwidth between partners are optimal"* [3]. The key is to make interfaces as simple as possible to maximize compatibility among models at the SoS level. Decomposing the Cohort Capstone team into interdependent tiger teams was essential for the engineering process, and ultimately, project success.
- <u>Heuristic 2</u>: *"Estimate using multiple methods" (analogy, parametric, etc.)"* [4]. Use of a number of modeling tools, varying from high level conceptual design and multidisciplinary decision analysis, to detailed, domain-specific solutions. AnyLogic, SysML, Microsoft Excel, Flowmaster, JMP and Phoenix Integration ModelCenter software packages were used to quantify uncertainty and determine areas for decision makers to make trade-offs. While this toolset was later down selected to more cost-effective and readily available software solutions in order to share the trade space tools with the customer, the analysis team leveraged as many options as possible upfront.
- <u>Heuristic 3:</u> *"Models can't replace decision makers"* [5]. Models are not reality, but instead act to help illustrate and simulate specific areas of the project. Therefore, models cannot replace the insight or funding constraints of decision makers. Their development helps the Aquarium decision makers to properly allocate funds to technology alternatives, understand uncertainty and ensure primary goals/milestones are communicated across the SoS spectrum.

Based on the heuristics above, the authors propose that the following attributes in defining a decision support process includes:

- A ***multi-layer system*** analysis – allowing technical specialists and decision makers to use a common, interlinked environment so as to collaborate in reaching a solution to a common problem. For example, a complex technical problem may generate different solutions across different "what if" technical scenarios (i.e. what if flow loss was reduced by 6%). These what-if solutions could be used as inputs to an accessible cost model that a decision maker could use to perform trade off scenarios and share findings. This process promotes transparency across stakeholders.
- A ***concurrent environment*** allowing for visualizing of "black box" driven results (i.e., push the "optimize button" and wait for the result) and user driven sampling and filtering exercises. This allows the user to compare their results for consensus building but also allowing different views of the optimization space (e.g., suppose the entire design space is too large to reasonably plot). The user can then enhance the running speed using fast greedy algorithms to keep the user involved in a more collaborative and knowledge driven approach.
- An ***Open Source environment*** and low-cost, highly available software integration approach that is cost effective and uses the immense wealth of smart libraries designed across the world.

4. Establishing a System of Systems Modeling Framework

For the current prototype, the team proposed a large set of processes, integrated some of these into the current analysis, and provided a roadmap for further integration. The current prototype uses:

- A single Excel spreadsheet hosting several dashboards that represent each technical system's design space (Greedy Algorithm and cost models) and connects the individual Excel dashboards into a System of System higher level visualization. Excel was chosen because everyone in the team (client included) is versed in its use. Additionally, VBA provided the platform to implement sampling to drive the greedy algorithms process across the dashboards.
- An Open Source optimization environment in Python – using a framework, such as OpenMDAO, which can provide access to complex workflow designs (sharing data across system designs) and implements optimization with uncertainties. The Python environment also has become widely used for data analysis. Since there are open source solutions for converting to and from UML and Python, the implementation could be extended to help manage the models and relationships in UML/SysML.

The framework to integrate the Python optimization with the Excel dashboards was created to allow the ability of calling Python code from VBA (using libraries such as "xlwings") which provides a roadmap for future integration. Additional requirements may be added later, such as allowing for collaboration via the web so that true concurrent analysis can be run across remote users. The current solution focuses on quick turnaround for a specific problem in the absence of such a user friendly but highly complex framework. The prototype was built in 10 weeks and was used to solve a complex SoS problem without access to Software Developers or buying expensive optimization and visualization software (beyond the simulation and modeling phase which required commercial physics based software to be used within this timeframe). The following diagram depicts the process of integrating each subsystem model into the following areas:

1. SoS level model and data
2. User Visual Interface
3. Optimization
4. Future integration improvement.

SoS modeling was organized into the staged approach shown in Figure 3. In this figure, the generic solution was broken into four stages identified by the red boxes and numbered according to the order in which each phase was addressed.

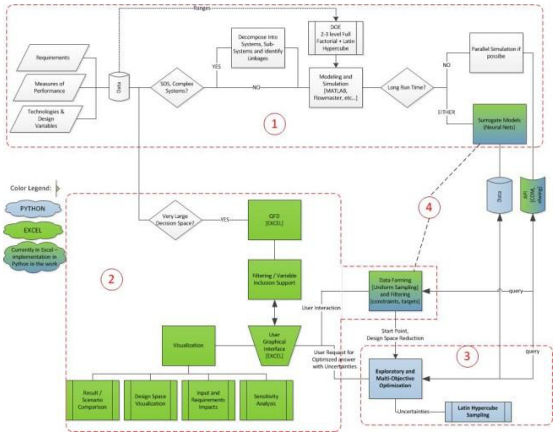

Figure 3. SoS Software Modeling Framework Staged Approach (Stages 1-4)

4.1. Stage 1 - System & Subsystem Individual Modeling and Simulation (M&S), Response Surface Modeling

Systems within the Ocean Voyager were modelled and simulated using a diverse set of software tools. Different tools are better suited for certain systems or system attributes. The software used to create the base data set are:

- MATLAB/Simulink: Renewable systems (solar, wind)
- Flowmaster: Hydraulic systems (PSK, Sand Filters loops)
- Microsoft Excel: Cost Models to include SoS level investigation and individual dashboard designs
- AnyLogic: Pedestrian / kinetic energy panel simulation

Input and output variables were selected so as to provide design specification changes to sub-systems based on common and linked measures of performance (energy demand and energy generation). For example, the hydraulic system for the Ocean Voyager was divided into two loops: "PSK" (protein skimmers) loop and "Sand Filter" loops (which includes sand filters, ozone contact, denitrification and aeration towers). Measured data for flow rate and pressure were used to correlate and create two baseline energy demand models in Flowmaster. These demand models were parameterized to support a design of experiments (DOE) analysis over the following design and control variables:

- Pump Design Rated Flow, Speed, Efficiency and Head
- Pump Control Speed
- Components (PSK, Sand, Ozone…) flow loss Coefficients

The DOE matrix was designed to generate data for the creation of surrogate response surface models (RSM) accessible to both the Excel Dashboard and the Python optimization processes. RSMs replace CPU intensive physics based models with faster than real time models with minimal loss in accuracy. In order to obtain good fit, the DOE space was sampled using 3-level full factorial (corners + mid-points) and a space filling Latin Hypercube (around 400 runs for PSK and 12,000 for the more complicated Sand Filter loop). The resulting data formed a surrogate model for each output versus input relationship.

4.2. Stage 2 - Excel dashboard, visualization and data farming concepts

For business and non-simulation expert level decision makers, a set of Excel dashboards were created around the individual systems. The systems represented the Aquarium system of system (SoS) architecture to allow exploration of the design trade space using the "data farming" methodology. Using MS Excel is beneficial because high-level decision makers already possess the skills to manipulate and understand MS Excel functions. The MS Excel approach also provides visual plots of the analysis and that is much better than looking at a large dataset of numbers. The overall goal of the dashboard approach was to provide an integrated environment for stakeholders to rapidly understand the solution space and the impacts of requirements on it. This enables real time decision making around complex SoS architectures and provides a SoS user interface that allows for added/editable design parameters to be incorporated by future users. The methodology for creating the dashboards has been outlined in the following sections.

4.2.1. Response Surface Model Integration

The RSM formulae outlined in stage 1 are imported into Excel. The dashboard contains the same design variables with the same respective ranges as those used to generate the RSM formulae. This allows the Excel dashboard to rapidly calculate the outputs based on the design variables. The same RSM formulae can be used in Python so as to integrate into optimization with OpenMDAO.

4.2.2. Data Farming

The "data farming" algorithm samples the RSM (using uniform distribution sampling of the design variables). This allows users to study the problem using a visual representation of the design space. The VBA "data farming" algorithm could be augmented or replaced by the DOE driver capability in OpenMDAO if the two were integrated, which would provide the capability to perform Latin Hypercube sampling. The user can then set constraints on the solution space (flow constraints for example) and visualize which design variable satisfies those constraints. Ranking the feasible solution for lowest power consumption provides the user with an understanding of how the system behaves. Optimization is then used to refine if necessary and to check the solution. Other advantages of the Neural Networks approach is the prohibitive cost of the CAE software and the difficulty of integrating them directly into the decision support tool. Another key factor is that software such as Flowmaster and AnyLogic have a significant learning curve which decision makers do not need to worry about. However, the simulation expert needs to choose the DOE input variables' range that will cover the range of analysis as well as generate enough runs to produce quality RSMs that the end user wishes to undertake.

4.2.3. Dashboard Development

Individual dashboards were built for the solar, wind, hydraulic, lighting, and pedestrian systems. Design (input) variables such as turbine blade length, solar array surface area, and efficiency were used to investigate utility (output). The dashboards permit users to investigate annual cost savings, annual power generation, and ROI based on user constraint entries such as maximum investment budget and minimum annual power produced and design variables. The hydraulic, solar, pedestrian, lighting, and wind dashboards were then integrated into an SoS dashboard to provide the user with a tool that can select an evolutionary, mixed architecture and look at multiple design variables that factor into cost and power. The dashboard for the GAI system of systems is shown in Figure 4. User input cells are highlighted in grey (shown in top middle of figure). The output cells are highlighted in blue. At the top left, the user can select the number of iterations to run the Monte Carlo simulation. The user can also adjust the design variables as checked or unchecked (Boolean) during the run of the design space. Additionally, the user can adjust the design variables with scroll bars. In the middle of the page, the user can select Hydraulics and Demand scenarios to simulate three different demand levels. The user also can input constraints for the maximum cost, Return on Investment (ROI), and energy savings. The output plots show the sub model design space and the SoS performance space for the ROI vs. cost and energy generated (kWh) vs cost.

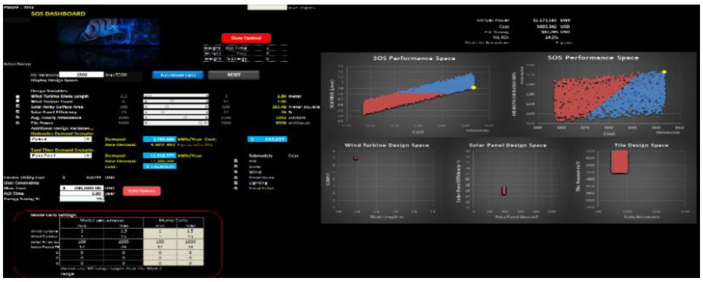

Figure 4. Example GAI System of Systems (SoS) Dashboard

4.3. Stage 3 - OpenMDAO and Utilization of Python wrappers

OpenMDAO was chosen as a low-cost/free solution for design optimization comparable to Phoenix ModelCenter. The concept of design optimization is based around decision theory which results in a design decision being made from design alternatives with outcomes based on uncertainty probabilities. This allows the selection of a utility parameter and corresponding equation to optimize design alternatives through analyzing uncertainties and performing DoE. OpenMDAO is an open source optimization framework written in Python that serves as a common platform software package that connects other individual software applications together. The open source aspect of OpenMDAO permits low-cost collaboration for users compared to optimization software packages that cost approximately $30K. Team AquaTech generated a framework in OpenMDAO for both GAI and future GIT PMASE cohorts. The framework included the following system models (Pedestrian, Solar, Wind, Ocean Voyager Hydraulics) incorporated into a higher level Georgia Aquarium Energy Model to investigate the design space and determine the optimal SoS solution. The model setup documentation created in Sphinx 1.2.2 can be referenced in Figure 5. Sphinx is an application that easily can create automated Python documentation that can be reviewed in an html repository. OpenMDAO also served to collectively investigate uncertainties at the energy model domain level for the Aquarium. The Uncertainties Driver documentation was also set up in Sphinx (Figure 6). In future iterations the bulk of the VBA scripts will be transferred into Python using the xlwing open source library.

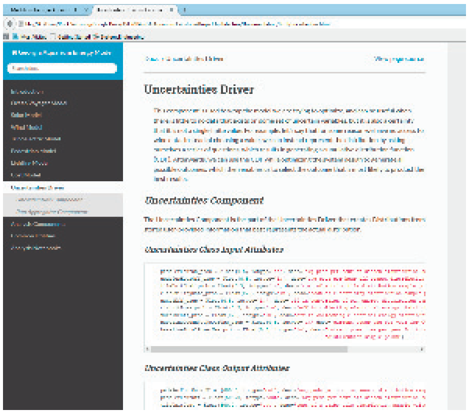

Figure 5. Uncertainties Driver Component

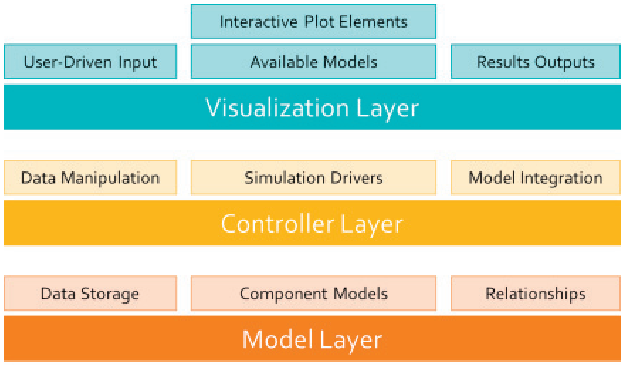

Figure 6. Decision Making Framework

4.4. Stage 4 - Integration of Python and VBA

The use of Python would separate the visualization component from the modeling component. The result is that Excel visualization or any other visualization tool would have much less impact on the models and the integrating elements. Therefore, you could extend either the models or visualizations and the resulting changes on each would be minimized. The flexibility and quick turnaround of something like Excel dashboards are advantages that enabled this type of work to be possible across many different problem statements, while providing the platform to prototype and iterate. Excel is powerful for creating visualizations quickly, but would be less robust in the long term if the experience was to be generalized and provided to a larger user base. If that is the case, then there are web-based visualization libraries that could be leveraged (Bokeh, Crossfilter.js, NVD3.js) as part of a web framework.

5. Recommendations

Use of the SoS trade space analysis framework provides insight into system sensitivities. Analysis shall be conducted to quantitatively compare design choices on cost and power as well as system budgets such as hydraulics, solar, wind, and lighting. Performing analysis at the SoS level additionally allows a combination of the individual systems to promote evolutionary upgrades and creating optimal returns on investment. Comparison between the design alternatives using the proposed SoS modeling framework provides justification for the recommendations provided, leveraging the sensitivity effects. The models show GAI what they currently have today in a terms of system characteristics captured in a set of system models, but this could be extended to additional domains within the aquarium within the construct that was developed. The way that the OpenMDAO models were developed were as standalone units with discrete interfaces, and will provide a strong foundation to extend off of for additional or more detailed analysis.

6. Conclusion

The main purpose of this research was to assess the systems of systems engineering (SoSE) approach and architectural processes applied to the development and evolution of the Georgia Aquarium. Establishing a tiger team framework early in the lifecycle helped align resources to project scope to meet the rapid schedule. Additionally, decomposing the architecture allowed individual models, including software, to be constructed for in-depth analysis and then unified to analyze the Aquarium from a SoS perspective. The suggested framework creates a low-cost, editable open-source environment investigating energy efficiencies and technologies for the Aquarium from the SoS perspective. As time progresses, the proposed framework will also allow enhanced collaboration for team members to prevent bottlenecks in model development and analysis. After extensive analysis, the PMASE 2012 Team AquaTech concludes that the established goals of 5% energy savings can be made with an investment budget of $400K or less. Modeling and simulation results demonstrated that near term improvements employing Sand Filter and Protein Skimmer system upgrades as well as integrating LED lighting into the Aquarium architecture would meet the energy goal. The SoS analysis calculated a 17% savings for electricity and cost. Using the software, the estimated SoS improvements cost $385,000 with a ROI of 1.1 years.

Shortcomings: Autonomy leans towards independence resulting in independent partners and constituents acting as independent actors. Aquarium exhibits have their own sponsors, funding, needs, and roadmap which challenge a SoS perspective.

Key Takeaways:

- M&S tools such as MATLAB Simulink, Flowmaster, Excel VBA, SysML, OpenMDAO, and AnyLogic can perform powerful analysis for organizations such as GAI.
- Architecture framework defines boundaries and interfaces to reduce complexity.
- Creating/aligning individual architectures for the SoS, software, & organizational structure helps governance, reduces complexity, reduces costs, and increases situational awareness for the multiple sponsors.
- The integrated software SoS approach allows an integrated environment for stakeholder real time decision making around complex SoS architectures.

References

1. Georgia Aquarium 2012 Year End Summary. Available at http://www.georgiaaquarium.org/2012AnnualReport.pdf
Boardman, J. J., & Sauser, B. B. (2006). System of Systems - The meaning of. Proceedings 2006 IEEE/SMC International Conference On
2. System Of Systems Engineering, 2006(Proceedings 2006 IEEE/SMC International Conference on System of Systems Engineering), 118-123.
3. Gorod, A., Sauser, B., Boardman, J., & DiMario, M. (2009). 'Satisficing' system of systems using dynamic and static doctrines. International Journal of System Of Systems Engineering, 1(3), 347-366. doi:10.1504/IJSSE.2009.029909
4. Valerdi, R. Heuristics for Systems Engineering Cost Estimation. Proceedings of 14th Annual PSM Users' Group Conference. 2010. New Orleans, LA
5. Smith, K. Modeling, Systems, Teamwork, and Quality. University of Minnesota. 2013.
6. Maier, M. W. & Rechtin, E. The Art of Systems Architecting / Mark W. Maier, Eberhardt Rechtin. Boca Raton, FL: CRC Press, 2009.
7. Jamshidi, M. Systems of Systems Engineering. Principles and Applications. CRC Press. 2009