30th Congress of the International Council of the Aeronautical Sciences
DCC, Daejeon, Korea ; September 25-30, 2016

VIRTUAL EXPERIMENTATION AND VISUAL ANALYTICS IN SUPPORT OF SYSTEMS ENGINEERING AND SYSTEMS OF SYSTEMS ENGINEERING EFFORTS

Kelly Griendling*, Olivia J. Pinon*, and Dimitri N. Mavris*
***Aerospace Systems Design Laboratory (ASDL)**
School of Aerospace Engineering
Georgia Institute of Technology, Atlanta, GA, 30332-0150, U.S.A.

Abstract

This paper addresses and illustrates the benefits of leveraging virtual experimentation and visual analytics to support system and system of systems engineering practices and efforts. It presents some of the challenges faced by the system of systems community throughout the systems engineering process and discusses the importance of virtual experimentation in the engineering of complex systems and the necessity to integrate it with visual analytics capabilities. In particular, two cases studies are presented that illustrate how virtual experimentation and visual analytics can be leveraged to help speed up the systems engineering process and provide meaningful engineering insights to support decision making.

1 Motivation

1.1 The System Engineering Perspective and Its Challenges

As Keane and Nair [1, 2] noted, "Aerospace engineering design is one of the most complex activities carried by mankind". The complexity of today's aerospace systems has far outpaced the advancement of the methodologies, processes and tools in which they are procured for the past half-century [5]. This divergence of design methodology and complex system procurement has materialized in the form of inefficient and costly aerospace projects that often exhibit the customary design-test-build-redesign cycle. This in turns has led to programs and products that are over budget, over schedule, and do not meet customers' expectations [6, 7].

Today's complex systems are multidisciplinary in nature and designed by geographically-distributed teams. They are often systems for which no empirical data is available, prompting the need to generate, gather and analyze data [8]. Most complex systems also rely on radically new technologies (new materials, electronics, etc.). They are characterized by tightly coupled interactions between subsystems and subcomponents that often result in the existence of unknowns-unknowns and consequently the presence of unintended, emergent intra-system behaviors that are difficult to trace [7].

It is widely acknowledged that, in such instances, a holistic and systematic approach that focuses on "the intrinsic interrelations of the systems of interest components and their extrinsic relationships to the greater whole" [9] and supports the development of a life-cycle balanced system solution meeting customer requirements is needed [10]. Indeed, properly implementing a Systems Engineering (SE) approach to the design and development of complex aerospace systems brings many benefits. Such benefits include reduced design cost and reduced design cycle time due to better upfront planning and more streamlined integration. In addition, a more thorough exploration of potential design concepts in the earliest stages of the design process can lead to a better overall end-product that is more robust to changes in requirements.

These benefits are achieved through [8, 11]:

- The analysis and exploration of the design space to support and improve the designer's understanding of the problem.
- The identification of potentially feasible concepts.
- The assessment of the sensitivities of the outcomes to the assumptions.
- The evaluation and quantification of trade-offs between metrics and concepts to identify the "best" solution.
- The verification and validation that the solution meets the set of requirements established and that it is acceptable.

However, the Systems Engineering approach suffers from "the inherent assumption that the sum of the parts will be equivalent to the whole" [7]. In reality, as discussed by Bloebaum and McGowan [7], in the case of large complex systems and systems of systems (as discussed in Section 1.2), the coupling and interaction between disciplines and subsystems are such that they make it extremely difficult to assess the impact that one change in a part will have on another part or the system as a whole.

In addition to the complexity of aerospace systems, it is important to acknowledge that these systems are also becoming increasingly interoperable. The increase in distributed operations, coupled with today's information age, leads to situations in which no system operates in isolation. While this increased focus on the system of systems (SoS) can result in increased performance, efficiency, and safety, the engineering of the system of systems has continued to present engineers with new challenges during the system design process. If not designed to properly integrate within the larger SoS in which it will operate, the fielded system may incur suboptimal performance, expensive upgrades post-deployment, and/or an inability to fill it intended role. This in turn would result in an overall degradation in the performance of the overall system of systems.

The following section will present some of the characteristics that define a system of systems and some of the associated challenges identified by the systems engineering community.

1.2 The Systems-of-Systems Perspective and Its Challenges

In general, there is no consensus on a single, all-encompassing definition of a system of systems. The DoD Defense Acquisition Guidebook defines a system of systems as "A set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities." [12]. The INCOSE BKCASE defines SoS Engineering as "an opportunity for the systems engineering community to define the complex systems of the 21st Century. While systems engineering is a fairly established field, SoSE represents a challenge for the present systems engineers at the global level. In general, SoSE requires considerations beyond those usually associated with engineering, to include socio-technical and sometimes socio-economic phenomena." [13]. A third definition, presented by Jamshidi and repeated in the INCOSE SEBoK is "A SoS is an integration of a finite number of constituent systems which are independent and operable, and which are networked together for a period of time to achieve a certain higher goal." [14, 15]. Many other similar definitions exist.

While helpful in understanding that a system of systems is generally composed of multiple, interacting, independently useful systems, and generally presents a higher degree of complexity than an individual system, there is no clear boundary. A general list of defining characteristics of a system of systems is provided below. This list, which has been generated through compilation of multiple sources in literature, is not intended to be a

comprehensive list. Rather, systems that exhibit some subset of these characteristics are likely to require the application of a system of systems engineering approach.
Compared to a System, a SoS might [16-18]:

- Be large in scope.
- Have complex integration.
- Consist of constituent systems that are operationally and/or managerially independent.
- Include elements which are geographically distributed.
- Be subject to high degree of uncertainty and risk.
- Evolve continuously with elements of differing lifecycles.
- Lack a single management/acquisition entity and have a broader range of stakeholders.
- Have elements which are not designed to fit the whole, and which are integrated post-design and deployment.
- Exhibit emergent behaviors.
- Have ambiguous requirements and fuzzy boundaries.
- Include humans as an integral part of the SoS.

In general, the managerial and operational independence of components tend to be the characteristics most commonly used to categorize a system of systems [15].

It is worth noting that many system of systems tend to be cyber physical systems, and this trend is increasing. The European CPSoS project defines a cyber-physical system as “large complex physical systems that are interacting with a considerable number of distributed computing elements for monitoring, control and management which can exchange information between them and with human users.” [19]. In a recent call for proposals, the NSF defined cyber physical systems as “Cyber-physical systems (CPS) are engineered systems that are built from, and depend upon, the seamless integration of computational algorithms and physical components.” [20]. Several authors have suggested additional characteristics of cyber physical system of systems, including [19, 21, 22]:

- Partial autonomy of constituent systems.
- Dynamic reconfiguration of the overall system on different time scales.
- A need for concurrent programming.
- Very large scale integration of networked embedded systems.
- Data proliferation.

The characteristics and complexity of these cyber-physical systems make testing and evaluation very challenging. Traditional system prototyping methods can be prohibitively expensive and/or time-consuming in cases with high levels of system complexity, where there are large numbers of interacting systems, or when the behaviors cannot be tested by anything less complex than the system itself.

The types of systems being described by the characteristics of system of systems and cyber physical system of systems have many recognized engineering challenges, including [16, 18]:

- High complexity makes overall management of design and requirements challenging.
- Management can overshadow engineering, especially in cases with broad range of stakeholders and funding sources.
- The initial requirements are likely to be ambiguous, and do not always correspond to measurable objectives.
- Systems elements must operate independently and collaboratively, and are not necessarily designed with the overall goals of the SoS in mind.
- System boundaries are fuzzy and can cause confusion.
- Continuous evolution of system means that systems engineering is continuously ongoing.
- SoS has a significant focus on data, information, and resource flows between systems that may not necessarily have been designed to interface between each other.

- Testing, validation and verification of modeling and simulation environments can be very difficult due to the nebulous requirements, the sheer size and number of systems, and, often, a lack of real world testing data to support validation.
- Because of the continuous evolution of the systems and the different life-cycles between systems, analysis results can become obsolete quickly.

1.3 Preliminary Remarks

One common limiting factor in the pursuit of system engineering and system of systems engineering is a lack of large amounts of relevant data to base designs, analyses and decisions from. This has significant implications on the systems engineering process and the ability to synthesize information on the integrated SoS. From a system design standpoint, the systems being designed today are highly unconventional in nature, which makes the use of historical data and heuristics incompatible with the task at hand. From a system of systems standpoint, the traditional reliance on prototyping as a key step in the design process is not always possible in SoS, as the size and complexity can make prototyping costly and/or time-prohibitive.

With reduced access to data on real world performance and reduced ability to scale and test design prototypes, it is necessary to increase reliance on virtual experimentation to support the systems process for these SoS. However, the amount of data generated through virtual experimentation can rapidly become overwhelming. The analyst or decision maker, when faced with such a data overload problem, is limited in his ability to conduct any kind of trades, test hypotheses, explore the design space, and detect unexpected trends, behaviors, detail or relations, as the data sets cannot be visualized [23]. Consequently, he cannot fully comprehend the problem to be solved, or understand the behavior of the system under consideration. While unprocessed data does not hold any intrinsic value [23], it can result in missed opportunities for critical actions. To alleviate this problem, it is necessary to move away from static representations and visualizations and develop means that enable the interaction between information and the analysts, while simultaneously allowing them to integrate their background, expertise, and cognitive capabilities into the analytical process. The need to address these aspects has given rise to a multidisciplinary perspective named Visual Analytics.

Section 2 presents two key enablers to support the engineering of complex systems and our understanding of systems-of-systems: Virtual Experimentation (Section 2.1) and Visual Analytics (section 2.2). Section 2.3. briefly discusses the complementary nature of these two enablers. Section 3.1 illustrates how integrating Visual Analytics (VA) in the design of complex systems can help facilitate a Systems Engineering approach and the realization of the benefits and capabilities mentioned in Section 1.1. Section 3.2 discusses the application of virtual experimentation and visual analytics to develop an unmanned systems virtual testbed for Naval applications, summarizing the creation and use of such an environment. Finally, this paper concludes on the importance of visual analytics and virtual experimentation in supporting the systems engineering process for system of systems design and decision making.

2 Leveraging Virtual Experimentation and Visual Analytics to Alleviate SE and SoSE Challenges

2.1 Virtual Experimentation

Virtual experimentation is based on the premise that many of the traditional physical testing methods can be replaced by highly accurate, high-fidelity, fully validated simulations. Just like wind tunnels are a key engineering resource, so are the simulations that enable virtual experimentation. Made possible by the surge in computing technology, it is now possible to execute simulations in higher fidelity and in greater numbers than ever before. The result is that this virtualization can lead to cost and time improvements in the design process. It should be noted that virtual experimentation is not expected to fully replace physical experimentation, only to reduce the application of physical experimentation to the most critical regions. The virtual experimentation platform should be able to be used to help identify those regions in which physical testing and prototyping are required to sufficiently mitigate uncertainty and risk. It is expected that virtual experimentation is even more critical in the engineering of system of systems, where prototyping of a fully integrated SoS is extremely difficult or even impossible in many cases.

The realization of virtual experimentation means that the creation of reliable and accurate simulations of the SoS becomes a critical piece of the systems engineering process, and these simulations must be able to correctly predict performance for systems or scenarios which have never been created or observed in the real world. However, the creation of the modeling and simulation environments that enable virtual experimentation is not a trivial task, and must be executed carefully if they are to be developed in a time and cost-efficient manner.

Thinking about the process of modeling, there are several steps that are generally accepted as necessary to create a model. These include (1) the generation of a conceptual model to translate the real world into a simplified representation that can capture desired effects, (2) the translation of the conceptual model into a set of pseudocode that can be programmed, (3) the programming of the model and verification that the programmed model matches the intent of the conceptual model, (4) validation that the results are an acceptable approximation of the real world phenomena, and (5) execution of the model to discover trends. All of these steps are necessary to the virtual experimentation process.

For a system of systems, the conceptual model is often represented in the form of a set of architecture views or models that define the boundaries of the system of systems, identify the constituent systems and services, define their relations and interactions (both physical and virtual), and describe the goals and intended usage of the system of systems. Part of the architecting process for a virtual experimentation environment therefore requires a clear definition of the system of systems architectures and its intended performance, as this will become the blueprint for the software architecting of the experimentation environment.

There are several types of modeling and simulation that are commonly applied to system of systems. These include Markov Chains, Petri Nets, system dynamics modeling, discrete event simulation, and agent-based modeling. Markov Chains, Petri Nets, and system dynamics models are particularly useful in discovering general trend and behaviors related to the flow of information and/or resources, and can be used to easily characterize the impact of stochastic effects on these flows. While Discrete event simulations are commonly used in the modeling of activity or process flows for the system of systems, they are particularly useful in the analysis of logistics and supply chain operations. Agent based approaches provide a ground-up modeling framework, which can be used to understand the complexity caused by the interactions and independent decision-making of entities in the system of systems. Agent-based modeling is often the most holistic approach, but it comes with a penalty of computational cost and time as compared to the other approaches discussed here. No one modeling approach is correct for every problem, and therefore the applicability and pros and cons of various approaches should be considered carefully when selecting a modeling framework for the implementation of the virtual experimentation platform.

Throughout the development and at the conclusion of the model-building process, verification and validation should be performed and carefully documented. It is the desire that the virtual experimentation environment continue to evolve and grow with the system of systems such that analysis results can stay relevant through the life cycle of the system of systems. Therefore, the documentation of the verification and validation, as well as the boundaries of applicability of the environment, is of upmost importance. One of the biggest challenges of the creation of a virtual experimentation environment for a system of systems is the validation of the environment. It may be that an initial validation is done using a SME-based approach and further aspects of the environment are validated in parallel with the development and fielding of system of systems, such that analysis on future upgrades can be well calibrated to the performance of the fielded system.

Ultimately, the use of the modeling and simulation environment is the key enabler for the systems engineering process. Using well-structured experimentation practices, the virtual experimentation environment can become a key part of the systems engineering process. It can be used to perform design space exploration, analysis of alternatives, sensitivity analysis, and uncertainty quantification, as well as help identify emergent or unexpected behaviors. It does, however, require careful planning of experimentation, as the number of required cases can grow quickly with the dimensionality of the problem. Furthermore, large numbers of runs with potentially large number of repetitions (when stochastic effects are present) can lead to large resulting data sets, which will then need to be understood. As detailed in the following sections, Visual analytics is an important tool for the use of the virtual experimentation framework, as it allows users to understand the behaviors internal to the model and better understand and analyze the results. The following subsection introduces visual analytics, and discusses how it can be an enabler to system of systems engineering. The subsequent subsection addresses the complementary use of these two enablers to improve the overall systems engineering process.

2.2 Visual Analytics

Initially introduced in the National Visualization and Analytical Center (NVAC) 5-year Research and Development Agenda for Visual Analytics [24], Visual Analytics has since developed into a field of study that benefits many sectors, from homeland security to commerce, healthcare and engineering [25]. Recently defined as “new enabling and accessible analytic reasoning interactions supported by the combination of automated and visual analysis” [26], Visual Analytics builds on diverse research areas and disciplines grouped into three main components: interactive visualization, analytical reasoning, and computational analysis.

The process illustrated in Figure 1 integrates both the visually enabled reasoning process, as defined by Meyer et al. [2] and the visual analytics process [3, 4, 27]. This disciplined, iterative, and interactive process, which supports scientific reasoning by means of data and visual analytics techniques, is described by the four following main steps:

Data preprocessing: The goal of this step, as explained by Kasik et al., is to prepare the data for visual representation by “identifying higher-order characteristics in the data, such as relationships, trends, summaries, clusters and synopses” [28]. Pre-processing may include data cleaning, selection, integration, transformation, etc. [3, 28] and is usually achieved through the implementation of mathematical, statistical, and linguistic techniques. Once the data have been pre-processed and transformed into efficient data representations, hypotheses can be generated (H_S) and visualizations can be created (V_S).

Hypotheses generation: Hypotheses can be formulated after applying analytical and statistical methods to the data (H_S). Additionally, hypotheses can drive the type of visualization to be used (V_H) and, inversely, visualizations can help formulate new hypotheses (H_V).

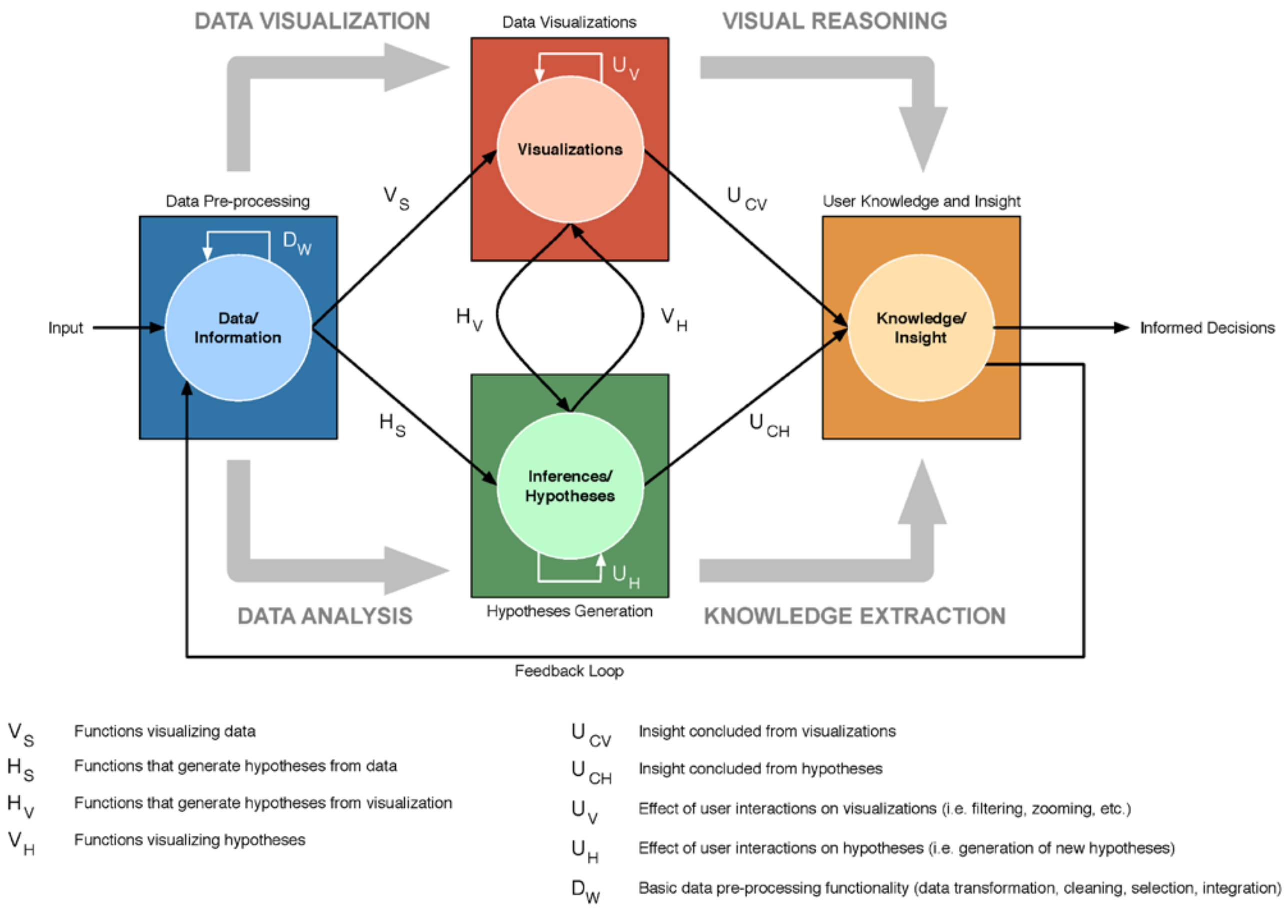


Fig. 1: The Visual Analytics and Enabled Reasoning Processes (Adapted from [3, 4])

Data visualization: Visualization is a means to extract and present relevant information from large volumes of generated or compiled data [27] in a format that enables reasoning and analysis, while allowing the user to navigate the overall space spanned by the data [28]. As explained by Kasik et al. [28], data representations obtained during the data pre-processing phase need to be further transformed (V_S) in order to provide the user with intelligible and effective visual representations.

User knowledge and insight: The knowledge or insight the user may gain from visualization (U_{CV}) and visual data exploration, in particular, is enabled through visual reasoning and depends on the level of information and interaction provided as well as his/her level of expertise and a priori knowledge. Based on his/her newly acquired knowledge and exploration objectives, the user may decide to obtain additional insight both on the data and its visualization. He/she may recompute the data and steer the analysis in a different direction (feedback loop) or he/she may dynamically interact with the visualization (U_V) through analytical means and techniques like brushing and linking, and/or panning and zooming, to focus on a different region or dimension of the data space. Exploring and investigating the data through a different or more focused angle can offer a new perspective on the problem, thus helping the user refine existing hypotheses and formulate new ones (U_H). These new hypotheses, through the process of knowledge extraction, may in turn contribute to increased insight (U_{CH}) and a better understanding of the problem.

Hence, through data manipulation, information visualization, and hypotheses generation and testing, the analysts is better equipped to learn about the problem and formulate informed decisions. In particular, when integrated into the design process and the System Engineering approach, this process allows the analyst and decision maker to [8]:

- Rapidly explore huge combinatorial spaces,
- Identify potentially feasible concepts or technology combinations,
- Formulate and test hypotheses,
- Steer the analysis by requesting additional data as needed (data farming),
- Integrate their background, expertise and cognitive capabilities into the analytical process,
- Understand and quantify trade-offs between metrics and concepts,
- Study correlations, explore trends and sensitivities,
- Provide interactive feedback to the visualization environment,
- Synthesize and share information,
- Investigate the design space in a highly visual, dynamic, interactive, transparent and collaborative environment, and
- Document and communicate findings and decisions

Overall, it contributes to helping teams get a better understanding of the system and the impact that design decisions have at higher levels.

2.3 The fusion of Virtual Experimentation and Visual Analytics

The application of virtual experimentation and visual analytics are complementary to each other. Visualization can be of assistance in almost every step of the virtual experimentation process. Architecture diagrams and other visual representations of the SoS are of assistance in the development of the conceptual model. UML and SysML are visual languages that can assist in the translation of the conceptual models to computer pseudocode, or even to actual code. The verification process can be greatly aided by a visualization of the behavior as a model is simulated. A comparison between the playback of a simulation run and the conceptual model can shorten the verification process and speed up debugging efforts. Visualization of model results can be used to aid in the overall validation as well as provide a means by which to synthesize large amounts of data into useful insights that can be used to support decision making in the engineering process.

In SoSE, it is possible that the validation piece of the process must rely on the opinion of subject matter experts (SMEs), where these experts confirm that the outcomes of the simulation make sense and that the conceptualization of the model is appropriate to the real world representation [29]. With SME opinion as a critical step in the modeling process, the ability for the SME to clearly understand the internal behaviors of the model is imperative. Furthermore, the complexity of the systems being represented can make the verification process extremely challenging. A well put together live, visual representation of the elements of the model and their behaviors can be a key enabler for both the verification and the validation process of the model, and can further be used to gain insights on the way certain behaviors and actions of constituent systems can drive outcomes. Visualizations of the live simulation can be key in identifying and understanding unexpected or emergent behaviors. An example of a visual front end that was developed for these purposes is presented as a case study in Section 3.2.

Once a virtual experimentation platform has been developed which has gained the confidence of its users through this verification and validation process, it can be employed to support a number of the functions of the systems engineering process. These functions include design space exploration, sensitivity analysis, risk analysis, and decision support, all of which are enabled and supported by Visual Analytics. As such, Visual Analytics is also a strong enabler in the practical use of virtual experimentation across the systems engineering process.

The following section presents two case studies that illustrate the benefits brought forward by Visual Analytics and Virtual Experimentation when applied in the contexts of system design and system-of-systems.

3 Case Studies

3.1 Visual Analytics in Support of a Digital Thread Approach to Manufacturing-influenced Conceptual Aircraft Design

As discussed in Section 1.1, the complexity of today's aerospace vehicles makes it extremely challenging to assess and visualize the impact that one change in a part will have on the system as a whole, or on the ability of the system to meet design requirements and other constraints. The coupling and interaction between design, manufacturing and production, in particular is one that is often difficult to capture.

With the recent shift to more composite aerostructures, historical regressions and cost estimating relationships used to predict the system's cost and producibility are no longer accurate. Hence, while this shift in material leads to important weight reductions, it brings a new set of issues and challenges to aircraft manufacturers. The ability to conduct multi-disciplinary trades in the early stages of design is thus crucial to better understand the interaction between a vehicle's physical design and its production system's performance (e.g. throughput, cost, efficiency, etc.). and eventually ensure its feasibility and profitability. For this multi-disciplinary design exercise to be successful requires all parties involved (vehicle design, production, and manufacturing engineers along with marketing and management personnel) to have access to data at the right time and at an appropriate level of detail. This can be achieved through the implementation of a digital thread approach that integrates aircraft performance considerations with production rate, manufacturing cost, and financial planning metrics into a parametric, visual trade-off environment. This environment, discussed in detail in [30, 31], allows designers and decision makers to:

- Parametrically and dynamically assess and visualize the impact that performance, manufacturing and production constraints have on the design space.
- Better understand the sensitivity of performance and production constraints on design and factory configuration variables.
- Gain insight into the efficiencies of the different manufacturing processes.
- Rapidly identify the critical path and potential problem areas in the production flow.
- Trade production rates, flow times, utilization rates, cost, and performance constraints.
- Rapidly identify design configurations and factory settings that lead to a desired throughput.

The Contour Profiler presented in Figure 2 enables the overlays of defined constraints onto two design variables of interest. The constraints and design variables can vary parametrically based on user inputs, opening or closing the feasible design space (white area) depending on their values. In this particular instance, the variables considered are high-level wing design variables (aspect ratio, wing area, etc.) and factory configurations variables (number of workers, tool sets, workstations, shifts, etc., for spar, ribs and skin lines). The data represented originates from the development and integration of models and codes into a multidisciplinary modeling and simulation environment [30], and the further use of surrogate modeling techniques for better integration within the visualization environment. An additional unique and critical feature of the contour profiler developed in the context of this research is its ability to concurrently visualize both traditional aircraft performance and production related constraints/requirements (number of wings produced/month). This allows the designer to observe the active constraints and identify the ones that prevent him/her the most from obtaining the largest feasible space possible and, consequently, from gaining the full benefits of the design concept. The superimposition of both performance and production constraints also

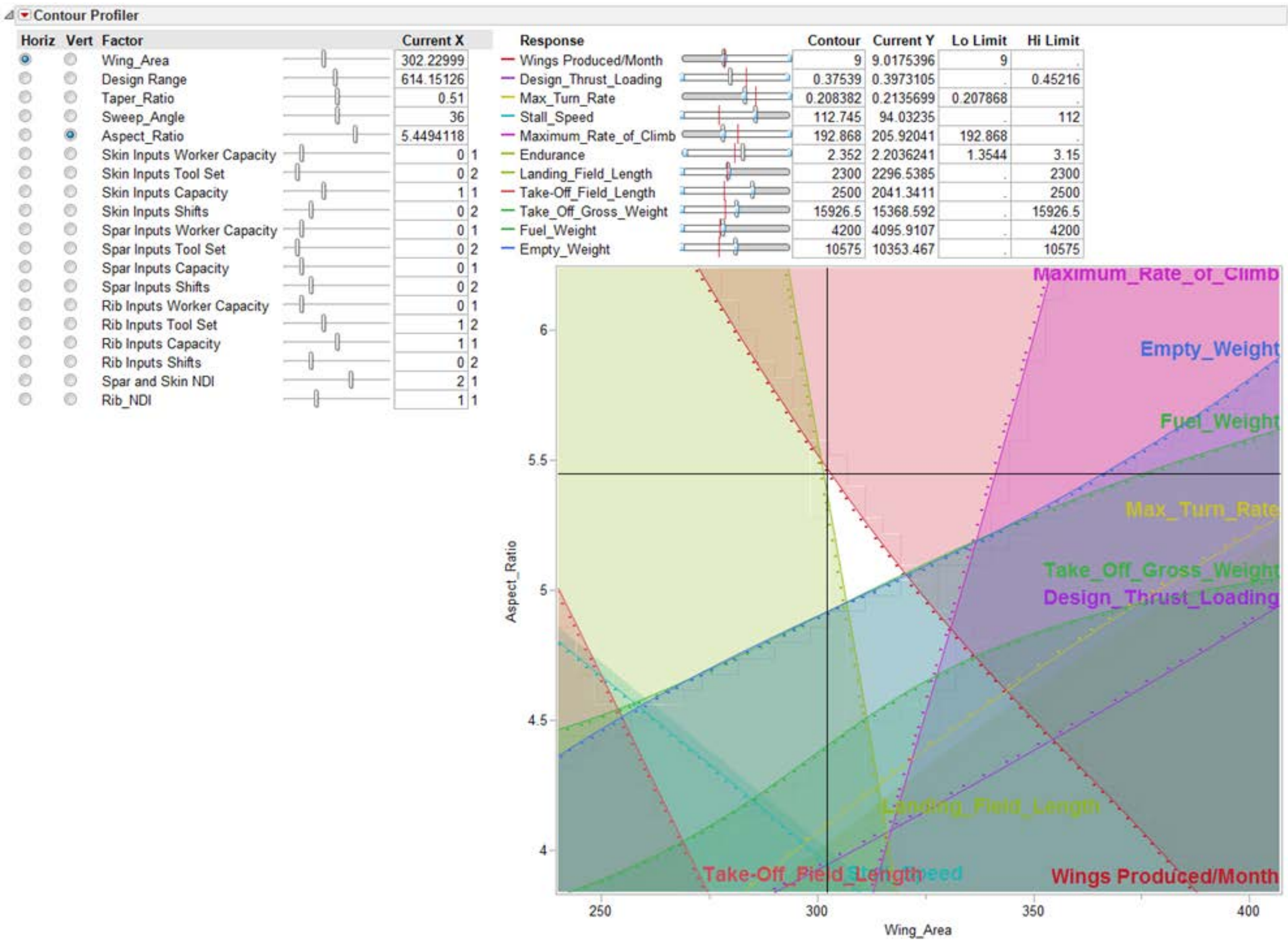


Fig. 2: Contour Profiler: Visualizing Performance and Production Constraints Parametrically and Simultaneously

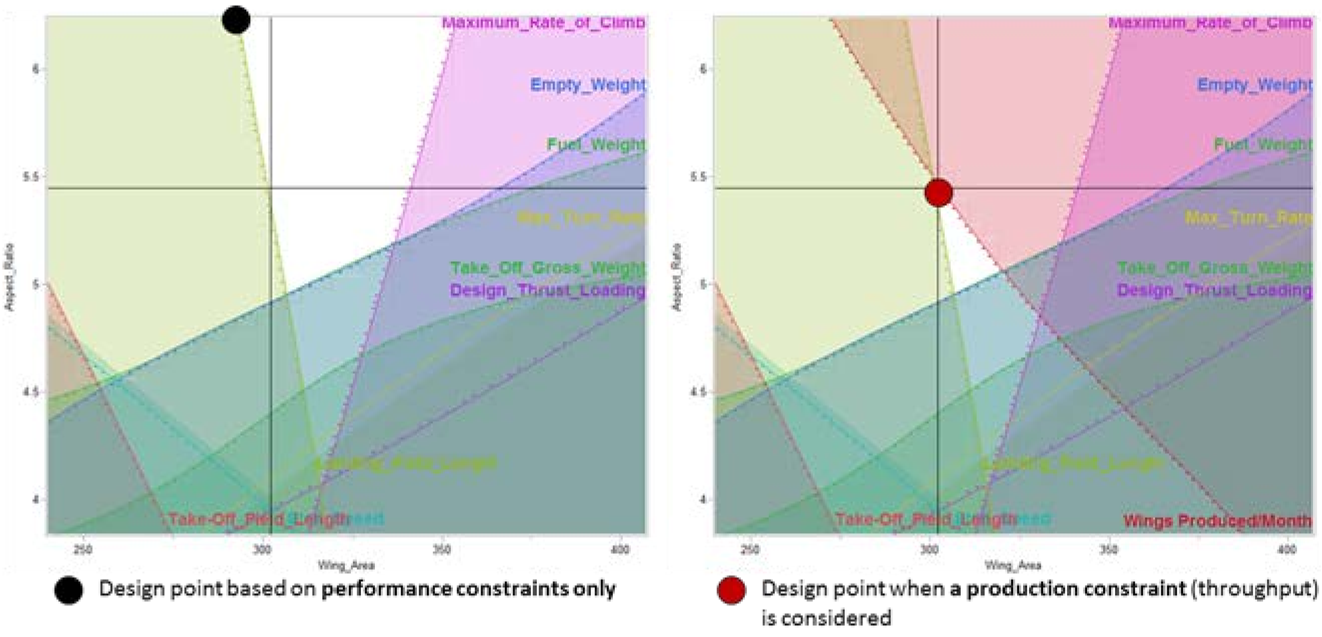


Fig. 3: Visualizing the Impact of a Production Constraint on the Feasible Design Space and the Selection of a Design

allows the designer to fully capture the impact that production may have on the choice of a design concept. In the example illustrated in Figure 3, the Contour Profiler can be used to rapidly assess and visualize the impact that adding a throughput constraint (e.g. the number of wings needed to be produced per month) has on the feasible design space and consequently the choice of a design. This capability helps ensure that the design chosen can be produced at the desired rate. Reciprocally, as illustrated in Figure 4, by superimposing production isoclines, the designer is rapidly informed of the impact that his/her chosen design point will have on throughput (e.g. wing produced per month).

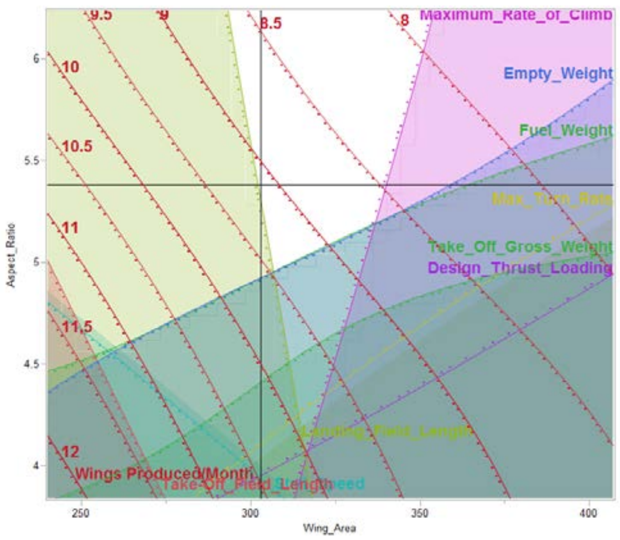


Fig. 4: Overlay of Production Isoclines

Wing area is a major driver for the dimensions of each component within the wingbox structure; scaling up the size of the wing increases the necessary thickness of each component, the lengths of the spars, ribs, skins, and stringers, and potentially increases the number of stringers. This results in longer manufacturing times and consequently lower throughput in terms of wings produced per month. While trends similar to this one are qualitatively understood, the aforementioned M&S and visualization capabilities provide the unique ability to quantify their impact, which in turn helps trace and facilitate discussions and compromises among the various stakeholders involved in the design.

3.2 A Virtual Experimentation Testbed: Unmanned Vehicles Collaborative Research Environment (UV-CORE)

As discussed in the previous sections, one outstanding challenge for system of systems comes in the creation, verification and validation, and use of complex, often non-physics-based simulation environments for virtual experimentation. These environments are rarely simple to validate against real world data, and are often being used to evaluate new concepts or configurations in areas of the design space where validation data simply does not exist. In these cases, verification and validation of the simulation requires a non-traditional approach. Furthermore, because these simulations are often executing complex behavioral models, it is often difficult to understand the chain of events that occurred in simulation leading to a particular outcome, or even debug and interpret simulation results. A traditional method of verification involves checking interim calculations, but this becomes more challenging when simulations involve stochastic events or agent intelligence in which units of the simulation have decision making algorithms and/or learning algorithms embedded in the simulation. In these scenarios, it is often helpful to have a way to "watch" the execution of a simulation case, in the same way a traffic controller would watch the behaviors of aircraft operating in the airspace.

An example of the development of a virtual experimentation testbed is presented here, using the Unmanned Vehicles Collaborative Research Environment (UV-CORE) developed at Georgia Tech and presented in [32]. UV-CORE was developed as a mission planning and research tool for aiding in experimentation of potential unmanned systems applications to a variety of Navy missions. It takes in a user-specified set of unmanned vehicles (including their vehicle performance, sensor payloads, path planning algorithms, communications hardware configuration, etc), a user-specified enemy configuration (including their vehicles, tactics, fleet size, etc), a user-specified set of operating concepts for friendly forces, the chain of command and information sharing structure for the vehicles, and the location, terrain, and atmospheric data. It then executes an agent-based simulation to predict the outcome of the mission.

Because there are stochastic effects modeled, each simulation case is run with repetitions to produce a distribution of the likelihood of different mission outcomes. For example, in the anti-piracy scenario, the user would specify the quantity, speed, and aggressiveness of pirates, the set of unmanned vehicles being deployed to combat the piracy, and the functions, actions, and decision logic for those vehicles. The simulation would produce distributions of measure of mission success for that mission, such as a distribution of successful interceptions of piracy attacks and the distribution of the number unsuccessful interceptions of piracy attacks, as well as distributions on other metrics of interest such as fuel burn, distance travelled, agents lost, etc.

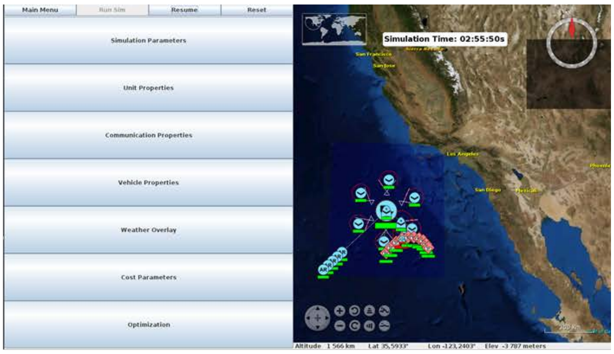
This simulation is an example of a case that presents many of the challenges described above. The Navy is exploring the possibility of using unmanned vehicles for these types of mission, but is not currently doing so. Therefore, there is no available data against which to validate the simulation outcomes. Furthermore, simulation results are the product of a large number of interactions and decision between agents in the simulation, all of which occur in the presence of stochastic effects. As a result, it is extremely challenging to verify and validate the model, or even to clearly trace how a particular execution of the simulation resulted in a particular outcome. Furthermore, the code base is very large, as it is attempting to provide the user with as much freedom as possible in defining scenarios to be explored. In order to help alleviate these challenges, a visual front end was created for the simulation that can be used to allow a subject matter expert to watch the simulation play out, understand the interactions occurring between the agents, understand what decisions are being made at what time and for what reason, and identify any unexpected behaviors to determine their cause.

In this particular case, it was logical to use a map of the area as the underlying basis for the visualization, as the relative positions of the systems were a key factor in their ability to successfully complete mission functions. Further, this allowed for easy inclusion of geographically-based features that impacted the mission, such as visualization of weather overlays and terrain. Each type of systems was given a unique icon, which moves through the space as the simulation runs. When interactions occur between systems, such as attempts at communications, colored lines are used to show the nature of the interactions and whether those were successful. In addition, agents that have been lost are frozen at the spot where they have been lost, and colored to indicate the loss. This allows a user to easily see how the mission and scenario evolve over time, and track the status of all agents. A feature was also added that allows the user to watch the trace of the agents' movements over time and help determine the goodness of the path planning algorithms. Metrics are tracked in real time in the upper corner of the play window, and the full set of input parameters is listed in a set of menus on the sidebar. All of these are interactive, such that all inputs can be manipulated by the user directly from the visualization window and a new run can be executed without ever leaving the visual environment. This feature is important because it allows two cases of interest to be compared back-to-back using identical visualization settings. An example of this interface and its layout is shown in Figure 5.

During the verification phase, this visualization environment proved to be invaluable in quickly identifying and fixing errors in the simulation. In one case, a scenario had been set up for an underwater search and engage mission in which friendly forces were believed to be significantly more capable than enemy forces. Upon execution of the simulation, however, the opposite was observed to occur; in no cases were the friendly forces successful in engaging the enemy, despite quickly locating and successfully tracking the enemy in almost every case. Simply watching the simulation play out in the visualization environment allowed for the rapid identification of the problem; the velocity of the torpedoes was being inadvertently maintained at 0 upon firing due to a typo during the code development. This could easily be seen in the visualization. When the torpedo was fired, the icon remained in place rather than moving along its expected trajectory. This allowed the developers to quickly identify the exact line of code which contained the error, and the problem was resolved within minutes.

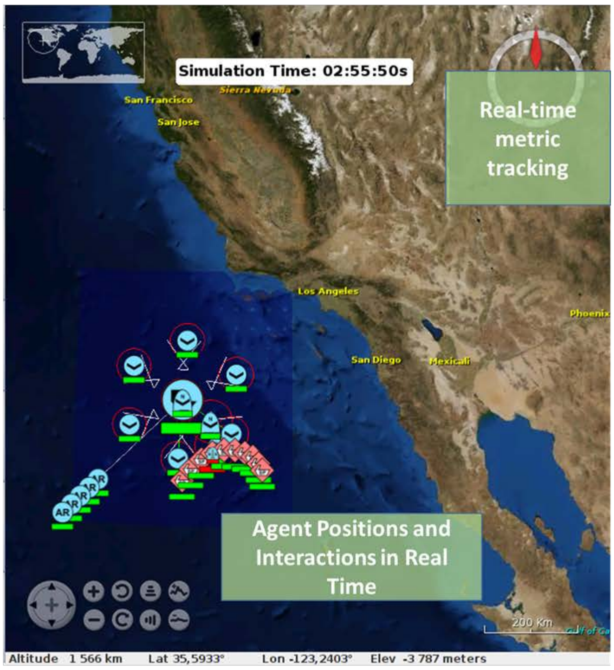
Without the presence of visualization, this very simple issue would have been very difficult to identify and track. This is only one example among many of how the use of visualization was able to accelerate the verification phase of the model development. The ability to do visual playbacks of the simulation helps to combat the previously mentioned challenges with respect to verification and validation. The behavior of the simulation can be easily verified against the developer's conceptual model, and SMEs opinion can be easily gathered by watching mission scenarios play out in the simulation to determine if the model is correctly representing the real world. In this case, this process was completed multiple times internally by the Georgia Tech team, after which the simulation was handed over to Naval partners for further evaluation.

As a second use of the environment, the visualization was implemented in an app form, such that it could be run on a touch table, as shown in Figure 6. This was done to allow for easy collaborative use of the simulation framework. In this case, users can gather around the table and watch mission scenarios play out, discussing behaviors which are observed as they happen. Experiments and scenarios of interest can be set up on the fly, and the group can collaborate on both the development of the scenario and the interpretation of the results. This allows the simulation to double as a mission planning tool once the SoS is deployed operationally, as discussed in [32]. Furthermore, this front end interface allows users to add new types of systems to the simulation on the fly, and to quickly update the simulation as the SoS evolves.



a) Simulation Visualization

b) Setup of Interface



c) Example of Playback Features

Fig. 5: Example of Visual Interface for Simulation

This helps to combat the challenges of the continuous evolution of the system of systems, and provides a way to evaluate potential SoS upgrades or modifications prior to implementation. As an example, the environment was used to perform a study on the application of collaborative path planning methods for underwater search. Several collaborative and non-collaborative approaches were compared, and a set of requirements for effective collaboration on underwater search was able to be developed. The reader is referred to [32] for a description of the use of the environment for a system of systems alternative analysis.

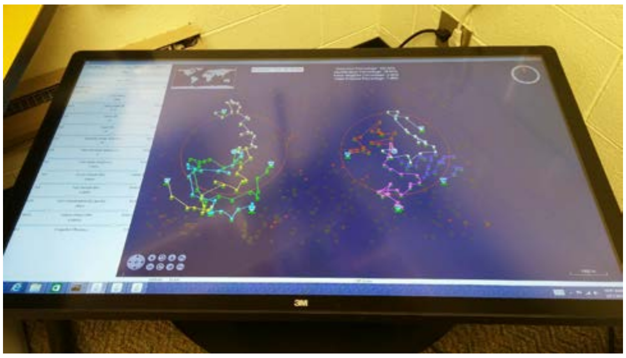


Fig. 6: UV-CORE Deployed on a Touch Table

These examples provide a subset of the ways that using a visual interface to virtual experimentation environment can help to perform SoSE functions more effectively and more efficiently, particularly during the development and use of a virtual experimentation testbed.

The particular virtual experimentation environment presented in this case study has been used to support a number of subsequent studies beyond its original application, supporting the idea that a virtual experimentation environment can be created which can be re-used across multiple analyses and scenarios. The UV-CORE environment was initially designed for evaluation and mission planning for the use of unmanned vehicles in surface warfare, but was later extended to include similar functionality for underwater search missions, as well as anti-submarine warfare missions [32]. It was then leveraged by another research team to perform assessment of collaborative control strategies and applicability of these strategies to maritime search and rescue. It is currently being used by a third research team as a platform for the assessment of the use of directed energy weapons from maritime platforms. The modular design of UV-CORE has allowed these extensions to be possible, but has also made it a desirable starting point for analyses of system of systems using similar components and/or system behaviors. Overall, UV-CORE presents a case in which a virtual experimentation environment was successfully developed, used, and extended to be a living testbed for a system of systems.

4 Concluding Remarks

This paper has presented a case for the application of virtual experimentation and visual analytics to the systems engineering process for complex systems and system of systems. Some of the key challenges tackled by this approach include the lack of sufficient data available during system design, difficulties in conducting traditional prototype-based, iterative testing and evaluation cycle for these systems, and difficulty managing the dimensionality of these problems. The primary argument put forth in this paper is that the complexity and size of these types of problems requires a new approach to systems engineering in order to manage cost and schedule concerns, and that the application of virtual experimentation and visual analytics are a necessary part of this transition. Two case studies were presented to demonstrate both the feasibility and value of applying such techniques during the systems engineering process. In both cases, it was demonstrated that a combination of a virtual experimentation platform with a visual analytics environment could speed up the systems engineering process and provide meaningful engineering insights to support decision making. The first case study focused on the value of using a visual analytics approach to support decision making in the systems engineering process. The second case study demonstrated the feasibility of creating and validating a system of systems virtual experimentation environment, and demonstrated the integration of visual analytics to benefit the simulation development and execution effort. Furthermore, the second case study demonstrated the feasibility of creating a virtual experimentation environment that is extensible across multiple problem areas, thus supporting the idea that these environments can become shared resources supporting multiple efforts. Overall, this paper recommends the broader application of these methods on complex systems and system of systems engineering problems.

5 References

1. Keane, A., Nair, P., *Computational approaches for aerospace design: the pursuit of excellence*. 2005: John Wiley & Sons, Ltd.
2. Meyer, J., Thomas, J., Diehl, S., Fisher, B., Keim, D., Laidlaw, D., Miksch, S., Mueller, K., Ribarsky, W., Preim, B., Ynnerman, A., *From Visualization to Visually Enabled Reasoning*. Scientific Visualization, 2010. **1**: p. 227-245.
3. Keim, D.A., Mansmann F., Schneidewind, J., Thomas, J., Ziegler, H., *Visual Analytics: Scope and Challenges*, in *Visual Data Mining*. 2008, Springer Berlin / Heidelberg. p. 76-90.
4. Riveiro, M., Falkman, G., Ziemke, T. *Visual Analytics for the Detection of ANomalous Maritime Behavior*. in *12th IEEE International Conference on Information Visualization (IV'08)*. 2008. London, UK: IEEE.
5. Felder, W.N., Collopy, P., *The elephant in the mist: What we don't know about the design, development, test and management of complex systems.* Journal of Aerospace Operations, 2012. **1**: p. 317-327.
6. Briceno, S.I., Pinon, O.J., Laughlin B., Mavris D. N. *Addressing Integration Challenges in the Design of Complex Aerospace Systems*. in *NAFEMS World Congress (NWC)*. 2013. Salzburg, Austria.
7. Bloebaum, C.L., McGowan, A-M. R. *The Design of Large-Scale Complex Engineered Systems: Present Challenges and Future Promise*. in *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSM*. 2012. Indianapolis, Indiana.
8. Mavris, D.N., Pinon O.J., Fullmer D. *Systems Design and Modeling: a Visual Analytics Approach*. in *27th Congress of the International Council of the Aeronautical Sciences (ICAS)*. 2010. Nice, France.
9. Vitiello, P.F., Kalawsky, R. S. *Visual analytics: A sensemaking framework for systems thinking in systems engineering*. in *2012 IEEE International Systems Conference (SysCon)*. 2012. Vancouver, BC.
10. Mavris, D.N., Pinon O.J., *A Systems Engineering Approach to Aircraft Design*, in *Encyclopedia of Aerospace Engineering*, R.B.a.W. Shyy, Editor. 2010, John Wiley & Sons.
11. Burge, S.E., *Systems Engineering: Using Systems Thinking to Design Better Aerospace Systems*, in *Encyclopedia of Aerospace Engineering*, R.B.a.W. Shyy, Editor. 2010, John Wiley & Sons, Ltd.
12. *Defense Acquisition Guidebook*, D.o. Defense, Editor. 2013.
13. INCOSE, *Body of Knowledge and Curriculum to Advance Systems Engineering* 2016.
14. Jamshidi, M., *Systems of Systems Engineering – Innovations for the 21st Century*. 2008, Hoboken, NJ, USA: Wiley and Sons.
15. INCOSE, *System of Systems (SoS)*, in *Systems Engineering Body of Knowledge*. 2016.
16. INCOSE, *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, 4th Edition*. 2015, San Diego, CA; USA: Wiley
17. Maier, M.W., *Architecting principles for systems-of-systems.* Systems Engineering, 1998. **1**(4): p. 267-284.

18. *Systems Engineering Guide for System of Systems*, S.a.S.E. Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Editor. 2008, ODUSD(A&T)SSE: Washington, DC.
19. CPSoS. *Towards a European Roadmap on Research and Innovation in Engineering and Management of Cyber-Physical Systems of Systems*. 2016 [cited 2016 6/30/16]; Available from: http://www.cpsos.eu/project/what-are-cyber-physical-systems-of-systems/.
20. Engineering, D.f.C.I.S., *Cyber-Physical Systems (CPS)* N.S. Foundation, Editor. 2016.
21. Lee, I., *Cyber Physical Systems: The Next Computing Revolution* 2009, Department of Computer and Information Science, School of Engineering and Applied Science, University of Pennsylvania
22. Lee, E.A., *Cyber Physical Systems: Design Challenges*. 2008, Electrical Engineering and Computer Sciences, University of California at Berkeley.
23. Keim, D.A., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., and Melançon, G., in *Information Visualization*, S.B. Heidelberg, Editor. 2008. p. 154-175.
24. Keim, D.A., Mansmann F., Thomas, J., *Visual Analytics: How Much Visualization and How Much Analytics.* ACM SIGKDD Explorations Newsletter, 2009. **11**(2): p. 5-8.
25. Thomas, J., Kielman, J., *Challenges for Visual Analytics.* Information Visualization, 2009. **8**(4): p. 309-314.
26. Thomas, J., Cook, K. A.,, *Illuminating the Path: The Research and Development for Visual Analytics*, I.C. Press, Editor. 2005.
27. van Wijk, J., *The Value of Visualization*, in *IEEE Visualization*. 2005. p. 79-86.
28. Kasik D. J., E., D., Lebanon G., Park, H., Pottenger, W. M., *Data transformations and representations for computation and visualization.* Information Visualization: Special Issue on the Foundations and Frontiers of Visual Analytic, 2009. **8**(4): p. 275-285.
29. Turner, A.J., *A methodology for the development of models for the simulation of non-observable systems*, in *Aerospace Engineering*. 2014, Georgia Institute of Technology.
30. Siedlak, D., Schmidt, T. M., Pinon, O. J., Mavris, D. N., *A Methodology for the Parametric Exploration of the Impact of Production Planning on the Early Stages of Design*, in *ASME 2014 International Manufacturing Science and Engineering Conference*. 2014: Detroit, MI, USA.
31. Siedlak, D., Pinon, O. J., Schlais, P. R., Schmidt, T. M., Mavris, D. N., *A Digital Thread Approach to Support Manufacturing-influenced Conceptual Aircraft Design.* Submitted to Research in Engineering Design, 2016.
32. Bays, M.P., Jim; Coker, Ayodeji; Cruzen, Christian; Braden, Julianne; Fouts, Cody; Griendling, Kelly; Mavris, Dimitri, *Modeling and Simulation of Unmanned Vehicle Sentry Missions to Assess Communications in a Maritime Environment.* Naval Engineers Journal, 2015. **127**(2): p. 105-110.

6 Contact Author Email Address

Contact author: Kelly Griendling
mailto: kelly.griendling@asdl.gatech.edu

7 Copyright Statement