31st Congress of the International Council
of the Aeronautical Sciences

Belo Horizonte, Brazil; September 09-14, 2018

A HOLISTIC ENGINEERING APPROACH FOR AERONAUTICAL PRODUCT DEVELOPMENT

Ingo Staack*, **Kristian Amadori***,**, **Christopher Jouannet***,**
*Linköping University, Linköping, Sweden
**Saab Aeronautics, Linköping, Sweden

Keywords: *techniques for SoS Engineering, morphological analysis, system of systems, systems engineering, decision support, heterogeneous domains, ontology, semantic model*

Abstract

Product development, especially in aerospace, has become more and more interconnected with its operational environment. In a constant changing world the operational environment will be subjected to changes during the life cycle of the product. The operational environment will be affected by not only technical and non-technical perturbations, but also economical, managerial and regulatory decisions, thus requiring a more global approach. One way to try tackling such complex and intertwined problem advocates studying the envisioned product or system in the context of System of Systems (SoS) engineering. SoS are all around us, probably in any field of engineering, ranging from integrated transport systems, public infrastructure systems to modern homes equipped with sensors and smart appliances; from cities filling with autonomous vehicle to defense systems.

Since also aerospace systems are certainly affected, this work will present a holistic approach to aerospace product development that tries spanning from needs to technology assessment. The proposed approach will be presented and analyzed and key enablers and future research directions will be highlighted from an interdisciplinary point of view. Consideration of the surrounding world will require to look beyond classical engineering disciplines.

1 INTRODUCTION

Typical envisioned usage (Concept of Operations (CONOPS)) of a complex (aerospace) system will certainly change during the life cycle of the product, due to changing, new, or even unforeseen external factors that largely influence the validity of the system. Traditional product development approaches based on optimizations with respect to a fixed set of requirements fail to provide robustness resilience in a constantly changing world. The problem becomes even worse when considering the long product life cycle that aerospace systems are designed for.

Furthermore, since today's aerospace products are often part of a larger integrated system, a System of Systems (SoS), it is important for the system manufacturer to be able to understand the relationships that lead from SoS needs, to required capabilities, to requirements on single constituent systems. Customers may have performed detailed SoS analysis to produce a specification document for a constituent system to be developed. But for the product manufacturer it is necessary to fully understand the customers' specifications and the underlying reasons. To engage requirement discussions and negotiations, suggesting trading certain requirements while demonstrating that the overall needs will still be met, the manufacturer must to be able to carry out similar analyses as the customer did.

Additionally, the manufacturer may desire trading some requirements to achieve a better alignment of the future product to its own business strategy, to the overall product portfolio, to technology development plans and to the currently available and future-planned in-house competence. Also, the same product may be developed for different customers at the same time, imposing a more holistic view, since particular needs may diverge and just producing a union of different requirement may lead to suboptimal solutions.

1.1 SoS Engineering

According to the definition by Maier [1], in this paper a SoS is assumed to posses five characteristic properties that sets it apart from conventional complex systems:

- Operational independence of the components
- Managerial independence of the components
- Geographic distribution of the components
- Emergent behavior of the system
- Evolutionary development of the system

The importance of the latter topic for product development is highlighted in the INCOSE Systems Engineering (SE) handbook [2], listing 31 product development processes for product life cycle engineering which may be required concurrently in a huge SoS with its underlying systems in different life cycle stages and parallel system upgrades. Gideon [3] proposed a taxonomy classifying every SoS by three type subsets. Applying Gideons taxonomy to large, complex aerospace SoS the following classification may apply: The SoSs are of *physical* domain type, most probable of a *dedicated* acquisition type and could be of any of the tree operational types, *directed*, *collaborative* or *chaotic*. The current paper will not try to address a particular SoSs within this classification, but rather try to understand the different phases needed to approach the development of a SoS regardless of the type or operational domain.

1.2 SoS research

Work performed by ADSL at Georgia Tech [4, 5] have proposed methodologies to tackle SoS in the context of defined scenarios and requirements. SEAri at MIT [6, 7] has chosen a different approach, focusing instead on epoch influences (see Sect. 3.6). Moving toward a larger scoop also implies that traditional engineering domains may not be sufficient. Needs may be dependent on socio-economical changes and therefore a broader set of domains must be understood and integrated.

From literature review such as presented by Axelsson [8] it can be noticed that SoS Engineering is not yet fully defined as a scientific discipline, and therefore no clear and holistic handbook/guidelines or best practices addressing the whole design process are existing. For this reason, this paper tries to offer a complete mapping of all steps within an overall SoS design process (as depicted in Fig. 1) including potential methods and tools that may support each phase. The goal of the paper is to outline a set of heuristics for SoS engineering and resilient design, but without proposing or developing deeper analytic methods.

1.3 The SoS Engineering Paradigm Shift

While conventional product development is primary a technical-focused process within established domains, modern approaches like DARPA FANG [9] propose instead to tackle product development based on cyber-physical simulations and model integration by means of some kind of a MDO framework (e.g. AGILE [10]). These approaches still belong to the mechanical engineering domain where huge progress has been made with respect not only to model implementation and modeling languages (like `Modelica, Catia, Python`, etc.), but also to available computational power and industry standards for model exchange and co-simulation such as the Functional Mock-up Interface (*FMI/FMU*) [11]. The primary concern of such solving frameworks is the early integration of physics-based models or methods of higher fidelity levels into the design process for design space exploration and optimization. Generally, the foundation of such frameworks relies on a parametric geometry model that serves as the central node to which domain-specific models are connected as functional extensions [12].

In order to add higher fidelity and include *non-mechanical engineering* domains, the field of study has to be extended to an interdisciplinary SE approach. This paradigm-shift adds several new domains and concepts to the design process of which the most important ones are addressed in Chapter 3. These extensions expand the design process not only up- and down-stream, but introduce also new domains and features to the design task such as business aspects, requirements and (stakeholder) needs handling, and technology selection including technology maturation planning.

2 HOLISTIC PRODUCT DEVELOPMENT IN THE SoS CONTEXT

An holistic approach to product development in the context of SoS is proposed and illustrated in Fig. 1. The goal of this phase-based process decomposition is to identify the main areas of interest in order to tie needs, capabilities, and system requirements in initial phases of product development. Five main phases have been identified as following:

- Needs and Boundary Conditions
- SoS Capabilities
- SoS Design Space

- Constituent Systems Design Space
- Sub-Systems Design Space

The phase breakdown is recurrent and the main liaison between them are described in Fig. 1. A brief overview of each phase and the coupling to the adjacent siblings is given in the following sections:

Phase 1 - Needs and Boundary Conditions
Within the product need analysis the needs related to the end-user needs are being analyzed. It is to highlight that these are not made equal to the stakeholders requirements. Typical high-level frames of interest in this phase are:

- Geopolitics, doctrine, laws and regulations
- Business cases
- Customer needs
- Threats, technologies
- Time frame (history, now, future), needs and boundary conditions

These analyses can be related to a fixed time frame or to different time frames, meaning that all of those inputs will be characterized by different levels of uncertainties and may vary within the different time frames. From a holistic perspective, those initial conditions and boundaries have to be varied in order to understand the main required capabilities in response to changing needs. The output is a set of different scenario-representing needs. These scenarios should be agnostic to any solution to understand the main capabilities required by the SoS.

Phase 2 - SoS Capabilities
SoS capabilities are defined by the scenario analyses. The underlying task is to figure out the impact of changes in the boundary conditions and the needs on the overall SoS capabilities. This analysis process leads to a balanced definition of the overall requirements on the SoS. Here, the capabilities design space is explored with the aim to understand it and to provide decision support for strategical choices. The output will provide main capabilities to be considered in the subsequent SoS design space exploration phase.

Phase 3 - SoS Design Space
With help of the architecture design space exploration, the SoS capabilities are transformed into a SoS design space containing all valid solutions that achieve the desired capabilities. Out of this pool, possible SoS concepts – including type and number of the constituent systems, collaboration and tactical models – are generated, responding to the different identified capabilities. Each SoS concept is represented by one entry in the SoS design space. This design space is then down-selected by benchmark processes to a short list of designs, each one of them made of a set of constituent systems. As an output, each constituent system will have a set of individual requirements.

Phase 4 - Constituent Systems Design Space
Based on the individual constituent systems requirements from the previous design phase, the design space for each constituent system is generated. Conceptual design of each constituent system is then performed based on the requirements provided from the SoS design space analyses. This phase will validate the feasibility of each envisioned constituent system of the short-listed design space.

Phase 5 - Sub-Systems Design Space
Sub-systems are the systems that constituent systems are made of. The sub-system design includes alternative architectures, system dimensioning and characterization, and compatibility and integration into complete constituent systems. Typically, the sub-system analyses will consist of domain-specific models for each disciplines within an constituent system. The process can be interpreted as a whole (classical) conceptual design phase for each system, preferably implemented in a highly integrated model-based systems engineering (MBSE) approach, enabling the analysis of a large amount of different architectures and configurations.

3 HOLISTIC DESIGN KEY ENABLER

Each process phase describe in Chapter 2 has its own challenges to be realized; some of them are more mature in methods and tools than others. A higher level of abstraction will be necessary to combine the different phases into one framework. This chapter presents a selection of different methods, research results and fundamental techniques that are identified by the authors as key enablers or available solutions to realize the envisioned process.

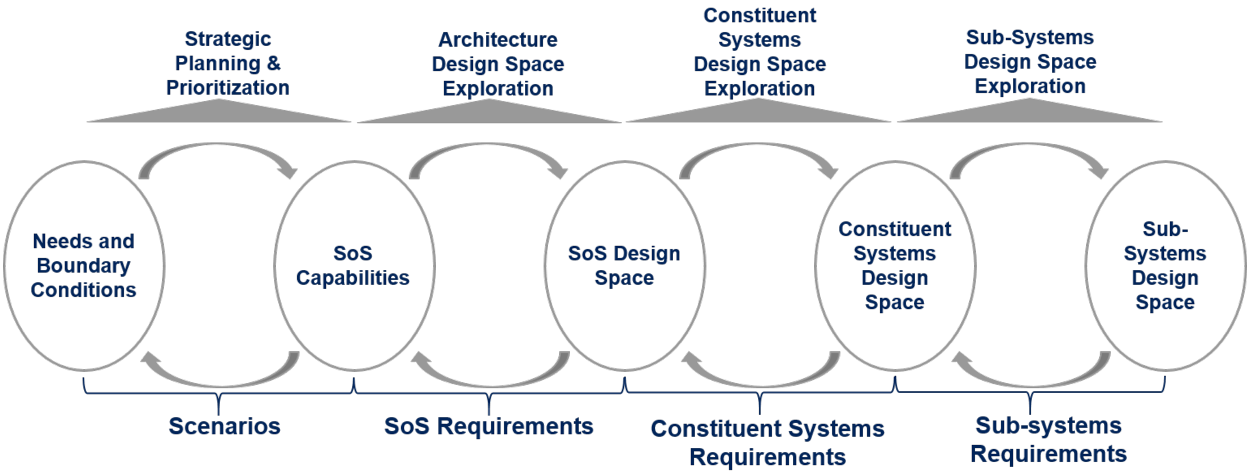

Fig. 1 Overview of the proposed System of Systems holistic design phases process.

3.1 Meta-Modelling and Common Language

In order to connect the different domains of the design phases illustrated in Fig. 1, a common language and semantic is required. Using ontology to describe a complex system or a complete SoS may be the way forward. A complete ontology (description) of the system of interest might theoretically represent a complete information input for a SoS design process beside the requirements. While examples of ontology for aeronautical applications can be found in publications [13, 14, 15], the usefulness of this approach for complex system/SoS engineering has yet to be shown. In a similar way, the DARPA FANG [9] and the DARPA AVM [16] projects focused on decreasing the product development time through component-based design and efficient cross-domain modeling. Large emphasis was put on the development of a model integration language, `CyPhyML` [17] and an semantic backplane `OpenMeta` [18, 19]. The selected tool for formal meta-modeling was `FORMULA` from Microsoft Research, a framework for formally specifying domain-specific languages (DSL) and model transformations [20].

From a mathematical point of view category and sheaf theory [21] could be the foundation for axiomatic description of the problem or the design space. This mathematical foundation seems promising and despite the fact that more applied research is needed to prove its usefulness, it has recently been acknowledge by DARPA as a corner stone for the DARPA FUN design [22]. Another approach to represent large and complex SoS have been applied by military organizations through the usage of System Modeling Language (SysML); creating an enterprise architecture approach to capture the information about the business in order to identify the processes and resources required to deliver the vision expressed in the strategy. Different variants of those architecture, depending on their origin, are available: MODAF/DODAF/NAF and IDEAS. The different architecture bear in common the different views:

- Strategic Viewpoint
- Operational Viewpoint
- Service Orientated Viewpoint
- Systems Viewpoint
- Acquisition Viewpoint
- Technical Viewpoint
- All Viewpoint

Those standards present the advantage of being based on a universal system modeling language, but have not yet been proven to be used within product development as a main backbone for the execution of model-driven design processes (unlike in the software engineering domain with e.g. executable UML (xtUML) models). Combinations of such framework- and service-oriented architectures may enable execution of SoS within its different viewpoints. Such a framework will serve as the link between views and models. The creation of domain-specific models however will still need to be performed in other frameworks/languages.

3.2 Matrix-based Approaches

Matrix-based information arrangement is a common and natural choice of representing (any type of) relationship between different entities. Introduced 1985 by Steward [23] for product (development) modeling it is usually denoted as Design Structure Matrix (DSM). In a certain arrangement, coupling the customer needs to the system characteristics, it is called the Quality Functional Deployment (QFD), also known as the House of Quality. One implementation strategy of a user-in-the-loop matrix-based product development process is the interactive reconfigurable matrix of alternatives (IRMA) process (see e.g. [24]). While the mathematics/logic relations in these (usually $2^{\text{-dim}}$) matrices are simple, the applied processes on theses matrices – namely sorting and clustering – are not; each of these processes represent a local optimization problem, fighting with the inherent problem of sheer unlimited number of combinations at already small-/medium-sized matrices [12].

Inherent problem of $2^{\text{-dim}}$ matrices in the $n^{\text{-dim}}$ design space is the fragmentation of clusters and acausal relationships[1]. Due to the break-up into a forward and a backward part of interconnected entities/modules, the matrix-based representation becomes difficult to read; this effects not only large and complex systems but occurs already at low complexities as low as triple or tetrahedron cluster formations [25].

Some single-domain DSM drawbacks can be mitigated by adding more domains to the DSM, extending the usual square $2^{\text{-dim}}$ (NxN) matrix into a composite $3^{\text{-dim}}$ $(NxNxD)$ matrix with D different domain matrices. However, due to the absence of a natural diagrammatic ($2^{\text{-dim}}$) representation of a multi-domain $3^{\text{-dim}}$ structure, graphical solutions to represent Multi-Domain DSM (MDDSM)s has to be found. A possible decomposition of a $3^{\text{-dim}}$ space into a $2^{\text{-dim}}$ space can be achieved by cascading the data and presenting the higher dimension within the cells of the first and second dimension. Abstraction can be achieved by the application of rating schemes e.g. by Pimmler and Eppinger [26], extended later by Helmer [27].

A significant difference between intra-system and intra-SoS relations is that most systems relationship within a SoS are communication channels for information exchange while physical system relationships often deal with the exchange of matter such as materials, fluids, energy, forces and heat. Consequentially, suitable modeling approaches (and tools) differ for both applications such as e.g. Unified Modeling Language (UML) and SysML for the former and `Modelica` or `Simulink` for the latter.

[1] Mathematical, logical and physical relationships such as matter exchange are usually acausal respectively bidirectional.

3.3 Relational / Graph-based

With the named disadvantages of matrix representations at hand, one solution to describe the system of interest is a graph network. With help of $3^{\text{-dim}}$ rendering, color schemes, arrows, and entities/cluster size, several domains can be represented human-understandable on a $2^{\text{-dim}}$ screen provided that the network entities have been arranged (and eventually clustered or sorted) with help of suitable layout (positioning) algorithms. Schaeffer [28] lists different mathematical approaches for graph clustering that can be applied for product modeling.

The advent of huge social networks and the associated data mining and analysis needs triggered the development of various tools, relational database systems, and data formats for graph structures (such as `Gephi`). Defining a relational network and editing can be made without any knowledge of the residue data unlike in a hierarchical databases approach such as the classical product tree structure. Every relation in the relational database/network is a resource-trait/aspect-resource triplet that establishes the relationship between two entities. These relational entities are data triples similar to the Resource Description Framework (RDF) triples used to model a ontology within the Semantic Web approach, originally invented by Bernes-Lee in 2001 [29] (see also Sect. 3.1).

3.4 Forcasting and Foresighting Methods

To define aerospace needs in future scenarios, forecasting or foresighting must be performed. The goal of forecasting is to provide prediction of highly probable future events, often based on extrapolation of known facts. In contrast, foresighting does not aim at predicting the future but rather serves *"... to explore the range of plausible futures that may emerge and to help identify assumptions and strategies that are robust in preparing for an uncertain future.[30]"*

Several different forcasting and foresighting methods exist and have been summarized by Kindvall [31]. The selective data collection process (typically executed by subject matters experts) will lead to recommendations of technologies and scenarios that have been identified as the most influential ones, see example from Silfverskiöld [32]. One inherent drawback of these methods is the subjective judgment that may affect the results. One key to use the findings from such methods would be to transform those scenarios and technology recommendations into models that can be part of the framework describe in Fig. 1. The application of foresighting within a framework for SoS engineering has been presented by Rhodes [33] and will be furthermore addressed in Sect. 3.6.

3.5 Value-driven and Robust Design

Value-driven design aims at shifting focus from the requirements only to understand and analyze the value for the customer brought into the SoS by different parts of the design. Underlying resectioning is to tie customer needs to the added value created by the different solutions. Methods proposed by Isaksson [34, 35] within aerospace applications show promising results and could be a valuable asset within the envisioned holistic product development process.

3.6 Epoch Analysis

Traditional SE tends to focus on meeting technical requirements. However, in a dynamic world, assumptions will probably change over time, affecting both technical and non-technical factors. One method to address those changes over time are epoch analyses proposed by Rhodes and Ross [6, 33]. Beesemyer [36] defines an epoch as *"...a period of time, defined by a fixed set of context and needs, which impacts the ultimate success of a system. A long-lived system may face a large number of epochs over its lifetime."*
The work performed by Rhodes SEAri group at MIT has shown the practicality of epoch analyses on various applications ranging from aerospace [37, 36] to maritime [38]. Application has mainly been on large complex systems, with some extensions to SoS [7, 39]. The authors of this paper feel confident that epoch analyses methods can be a key enabler for setting up the first phase in the proposed holistic development approach.

3.7 Data-driven Design and Tradespace Exploration

Tradespace exploration is not only a way of creating more design solutions than current methods. It is envisioned to be an interactive visual environment, enabling live *what-if* questioning to cover more criteria than commonly applied in early conceptual design phases. The goal is to provide resilient system solutions in a changing context and long-term perspective inherent to future aerospace SoSs. To perform such tradespace analyses, a data-driven approach is mandatory to enable a unremitting evaluation and analysis of alternatives. Data-driven methods rely on large computations with sensitivity analyses performed on all relevant variables. In contrast to current approaches where requirements are considers as the primary input to product development, the aim of tradespace exploration is to generate the system requirements [40]. Tradespace exploration techniques and diverse applications have been presented in large extend by the MIT SEAri group [41, 42]. The U.S. Department of Defense (DoD) funded recently the Engineered Resilient Systems (ERS) project [43] to explored more efficient ways for military acquisition. As a result of the ongoing effort the DoD want to leverage data-driven design as well.

3.8 Visual Analytics and Big Data

The authors recognize the need to incorporate big data handling coupled with efficient interactive visualization as a key capability. The different design spaces within each phase of the proposed holistic design process will lead to a very large set of data that needs to be managed and understood to support a well informed decision making. Georgia Tech has for a long time advocated using visual analytics as an assistive technology for decision support [44, 45] to render possible large SoS design space explorations and uncertainty quantifications. Also within military applications, visual analytics and big data are being identified as key enablers for efficient acquisition of military products in the future (see the previously named ERS project of the U.S. DoD). The Swedisch Defence Research Agency (FOI) published recently a comprehensive summary of the current research state of visual analytics methods [46].

3.9 Other Domains

Most of the identified SoS enabler in this chapter origin from engineering domains. However, to realize the envisioned holistic development approach, additional domains have to be investigated and understood to benchmark their impact and capabilities concerning the design space exploration. Some key thoughts are presented here and should not be seen as a definitive list but rather as an actual status of the authors knowledge.

Economic decision-making studies performed by the economist Thaler [47, 48] incorporating psychological realistic assumptions, limited rationality, social preferences, and lack of self-control of the stakeholders. These studies show that external factors have a large (non-rational) influence on decision making. Consequently, similar methods and assumption must be incorporate into the product development process, where customer preferences may certainly be influenced by similar factors.

The availability and recent progress of Artificial Intelligence (AI) can be an opportunity for decision support and large data analyses within the context of trade studies. It may also support better domain-specific understanding as well as help to identify advantageous and disadvantageous emergent cross-domain coupling effects. Further understanding of current research (from the authors' point of view, all with a engineering background) is needed to incorporate the non-engineering disciplines such as geopolitical modeling and assessment in future implementations. Application examples that may largely benefit from ongoing machine learning (AI) and natural language research are meta-modeling and socio-economical domain representations.

4 CURRENT LIMITATIONS AND THE WAY FORWARD

Paying heed of all domains within one holistic SoS approach seems today sheer infeasible due to the overwhelming complexity and the different modeling approaches within each field. One reason is the lack of a established *holistic SoS* research and education field with the consequence that existing solutions are biased by the research groups' background such as mechanical engineering, computer science, social psychology, mathematics, and so forth. With nowadays knowledge, it is not clear whether such a solution can be solved by a distributed (master-master) framework of different domain experts or whether a single master domain has to be found to take lead of the whole orchestra. May a uncontrolled symphony of different domain conductors lead to the wished outcome?

A central point in the implementation strategy has to be the decision of a *machines first* or *humans first* approach [49]. How much of the design process can and has to be understood by the involved persons? How can the output be actively influenced, how the operator integrated in the tool machinery? A pure AI-like behavior might be not acceptable due to sensitivity and traceability requirements needed for trade-off analysis. The (direct) use of training-based AI methods – also denoted as big data mining – such as Neural Networks (NN) may be limited for SoS due to the lack of relevant training data, although it appears appealing to make use of a NN-algorithm analyzing the ontology description of the system of interest. The absence of empirical data is also the critic of Axelsson [8] on most SoS publications. Here, inspiration from the software engineering domain should be applied which will presumably lead to widely accepted systematic methods and best practices within the SoS research community.

Are there further research fields SoS can be inspired of? In comparison to SoS research, other (classical) scientific fields are more matured and as a consequence converged toward international standards and widely accepted best practices. Close, large and complex collaborative work can for example be found in the field of experimental physic(s), medicine, economic sociology (irrational behavior) and climate change research. All of these stakeholders have in common to be either based upon on big data analyses or in need of complex/expensive/high-risk/long term experiments or models.

References

[1] Maier M W. Architecting principles for systems-of-systems. *Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998.

[2] Walden D D, Roedler G J, Forsberg K J, Hamelin R D and Shortell T M. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. 4th ed. International Council on Systems Engineering, John Wiley & Sons, 2015.

[3] Gideon J, Dagli C H and Miller A K. Taxonomy of systems-of-systems. *Conference on Systems Engineering Research*, pp. 356–363. Institute of Electrical and Electronics Engineers (IEEE), 2005.

[4] Mavris D N and Jimenez H. An evolution of morphological analysis applications in systems engineering. *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 972. AIAA, 2010.

[5] Roberts W, Griendling K, Gray A and Mavris D N. Unmanned vehicle collaboration research environment for maritime search and rescue. *30th Congress of the International Council of the Aeronautical Sciences*. ICAS, Daejeon, South Korea, 2016.

[6] Rhodes D H, Ross A M and Nightingale D J. Architecting the system of systems enterprise: Enabling constructs and methods from the field of engineering systems. *3rd Annual IEEE International Systems Conference*, pp. 190–195. Institute of Electrical and Electronics Engineers, 2009.

[7] Rhodes D H and Ross A M. Shaping socio-technical system innovation strategies using a five aspects taxonomy. *7th European Systems Engineering Conference (EuSEC)*. International Council on Systems Engineering (INCOSE), 2010.

[8] Axelsson J. A systematic mapping of the research literature on system-of-systems engineering. *10th System of Systems Engineering Conference (SoSE)*, pp. 18–23. 2015.

[9] Eremenko P. FANG: Fast, adaptable, next-generation ground vehicle. Tech. rep., Defense Advanced Research Projects Agency (DARPA), 2011.

[10] DLR. AGILE: Aicraft 3rd generation MDO for innovative collaboration of heterogeneous teams of experts. Horizon 2020-EU.3.4., Project: 636202, 2018.

[11] Modelica Association. FMI-standard: Functional mock-up interface, 2016. [Online; accessed 2016-04-14].
URL http://www.fmi-standard.org

[12] Staack I. *Aircraft Systems Conceptual Design*. Linköping studies in science and technology. dissertation no. 1805, Department of Management and Engineering (IEI), Linköping University, Sweden, 2016.

[13] Ast M, Glas M and Roehm T. Creating an ontology for aircraft design: An experience report about development process and the resulting ontology. *Deutscher Luft- und Raumfahrtkongress*, 301256, pp. 1–11. Deutsche Gesellschaft für Luft- und Raumfahrt-Lilienthal-Oberth eV, 2013.

[14] Reiss M, Moal M, Barnard Y, Ramu J P and Froger A. Using ontologies to conceptualize the aeronautic domain. *Proceedings of the International Conference on Human-Computer Interaction in Aeronautics*, p. 56–63. Cépaduès-Editions, 2006.

[15] Glas M. *Ontology-based Model Integration for the Conceptual Design of Aircraft*. Ph.D. thesis, Technical University Munich (TUM), 2013.

[16] de Weck O L. Fast adaptable next-generation ground vehicle challenge: Challenge analysis. DARPA # HR0011-13-C-0041 Phase 1 (FANG–1), Intelligent Action Inc., U.S., 2013.

[17] Sztipanovits J, Bapty T, Neema S, Koutsoukos X and Jackson E. Design tool chain for cyber-physical systems: Lessons learned. *Proceedings of the 52nd Annual Design Automation Conference*, DAC '15, pp. 81:1–81:6. ACM, New York, NY, USA, 2015.

[18] Simko G, Levendovszky T, Neema S, Jackson E, Bapty T, Porter J and Sztipanovits J. Foundation for model integration: Semantic backplane. *32nd Computers and Information in Engineering Conference*, vol. 2, pp. 1077–1086. American Society of Mechanical Engineers (ASME), 2012.

[19] Simko G, Lindecker D, Levendovszky T, Neema S and Sztipanovits J. Specification of cyber-physical components with formal semantics – integration and composition. *Model-Driven Engineering Languages and Systems*, pp. 471–487. Springer Berlin Heidelberg, 2013.

[20] Jackson E, Levendovszky T and Balasubramanian D. Reasoning about metamodeling with formal specifications and automatic proofs. *International Conference on Model Driven Engineering Languages and Systems (MODELS 2011)*. Association for Computing Machinery (ACM)/IEEE, 2011.

[21] Kashiwara M and Schapira P. *Categories and Sheaves*. Grundlehren der Mathematischen Wissenschaften. Springer, 2006.

[22] Darpa. Disruption opportunity special notice darpa-sn-17-71 fundamental design (fun design). Tech. rep., Defense Sciences Office (DSO), DARPA, 2017.

[23] Steward D V. The design structure system: A method for managing the design of complex systems. *IEEE Transactions on Engineering Management*, vol. EM-28, pp. 71–74. Institution of Electrical Engineers (IEEE), 1981.

[24] Engler W, Biltgen P T and Mavris D N. Concept selection using an interactive reconfigurable matrix of alternatives (IRMA). *45th AIAA Aerospace Sciences Meeting and Exhibit*, 1194. 2007.

[25] Yu T, Goldberg D E, Sastry K, Lima C F and Pelikan M. Dependency structure matrix, genetic algorithms, and effective recombination. *Evolutionary Computation*, vol. 17, pp. 595–626. MIT, 2009.

[26] Pimmler T U and Eppinger S D. Integration analysis of product decompositions. *ASME Conference on Design Theory and Methodology*, pp. 343–351. American Society of Mechanical Engineers (ASME), Minneapolis, MN, 1994.

[27] Helmer R, Yassine A A and Meier C. Systematic module and interface definition using component design structure matrix. *Journal of Engineering Design*, vol. 21, no. 6, pp. 647 – 675, 2010.

[28] Schaeffer S E. Graph clustering. *Computer Science Review, Elsevier Ltd*, vol. 1, no. 1, pp. 27–64, 2007.

[29] Berners-Lee T, Hendler J and Lassila O. The semantic web: A new form of web content that is mean-

ingful to computers will unleash a revolution of new possibilities. *Scientific American*, 2001.

[30] Kuosa T. *Towards strategic intelligence : foresight, intelligence, and policy-making*. Dynamic Futures, Vantaa, 2014.

[31] Kindvall G, Lindberg A, Trane C and Westman J. Exploring future technology development. Tech. rep., FOI, 2017.

[32] Silfverskiöld S, Liwång H, Hult G, Sivertun Å, Bull P, Sigholm J, Lundmark M, von Gerber C, Andersson K and Sturesson P. Technology forecast 2017 - military utility of future technologies. Tech. rep., Report from Seminars at the Swedish Defence University's (SEDU) Military-Technology Division, 2017.

[33] Ross A M and Rhodes D H. Using natural value-centric time scales for conceptualizing system timelines through epoch-era analysis. *INCOSE International Symposium 2008*, pp. 1–15. International Council on Systems Engineering, 2008.

[34] Bertoni M, Bertoni A and Isaksson O. Evoke: A value-driven concept selection method for early system design. *Journal of Systems Science and Systems Engineering*, vol. 27, no. 1, pp. 46–77, 2018.

[35] Isaksson O, Kossmann M, Bertoni M, Eres H, Monceaux A, Bertoni A, Wiseall S and Zhang X. Value-driven design : a methodology to link expectations to technical requirements in the extended enterprise. *23rd Annual International Symposium of the International Council on Systems Engineering*, vol. 1, pp. 171–187. International Council on Systems Engineering (INCOSE), 2013.

[36] Beesemyer C, Ross A M and Rhodes D H. Case studies of historical epoch shifts: Impacts on space systems and their responses. *AIAA Space 2012 Conference*, pp. 1–13. American Institute of Aeronautics and Astronautics, 2012.

[37] Rader A A, Ross A M and Fitzgerald M E. Multi-epoch analysis of a satellite constellation to identify value robust deployment across uncertain futures. *American Institute of Aeronautics and Astronautics*, pp. 1–16. AIAA, 2014.

[38] Gaspar H M and Erikstad S O. Handling temporal complexity in the design of non-transport ships using epoch-era analysis. *International Journal Maritime Engineering*, vol. 154. Transactions RINA, 2012.

[39] Parker V D, Ross A M and Rhodes D H. Program and portfolio affordability tradeoffs under uncertainty using epoch-era analysis. *INCOSE International Symposium*, vol. 26, no. 1, pp. 2391–2406, 2016.

[40] Rhodes D H and Ross A M. Esd 411: Concept design and tradespace exploration. Online coure material, 2014. [Online; accessed 2018-03-27]. URL http://seari.mit.edu/documents/presentations/ESD411Oct14_RossRhodes_MIT.pdf

[41] Ross A M, McManus H L, Rhodes D H and Hastings D E. Revisiting the tradespace exploration paradigm: Structuring the exploration process. *AIAA 2010 Space Conference*, pp. 1–14. American Institute of Aeronautics and Astronautics, 2010.

[42] Various. Example mate projects, 2008. SEAri Working Paper Series, WP-2008-5-2, Massachusetts Institute of Technology. URL http://seari.mit.edu/documents/working_papers/SEAri_WP-2008-5-2.pdf

[43] Holland J P. Engineering for resilience. *AIAA SciTech Conference*. American Institute of Aeronautics and Astronautics, 2016.

[44] Mavris D N, Balchanos M, Sung W and Pinon O J. A data mining and visual analytics perspective on sustainability-oriented infrastructure planning. *Data Mining and Big Data*, pp. 330–341. Springer International Publishing, Cham, 2016.

[45] Mavris D N, Pinon O J and Fullmer D J. Systems design and modeling: A visual analytics approach. *Proceedings of the 27th International Congress of the Aeronautical Sciences (ICAS)*. Nice, France, 2010.

[46] Jandel M, Bivall P, Hammar P, Johansson R, Kamrani F and Quas M J. Visual analytics: Perspectives on the field of interactive visualization. Tech. rep., FOI, FOI-R—4200—SE, Stockholm, 2016.

[47] Mullainathan S and Thaler R H. Behavioral economics. NBER Working Papers 7948, National Bureau of Economic Research, Inc, 2000.

[48] Barberis N and Thaler R. A survey of behavioral finance. NBER Working Papers 9222, National Bureau of Economic Research, Inc, 2002.

[49] INCOSE. First incose workshop on the evolution of human-systems integration. Tech. rep., Human-System Integration Working Group, Florida Institute of Technology, U.S., 2016.

Contact Author Email Address

ingo.staack@liu.se

Copyright Statement