Set-Based Design

Lead Authors: *Eric Specking, Gregory S. Parnell, and Ed Pohl*

Set-based design (SBD) is a complex design method that enables robust system design by 1) considering a large number of alternatives, 2) establishing feasibility before making decisions, and 3) using experts who design from their own perspectives and use the intersection between their individual sets to optimize a design (Singer, Doerry, and Buckley 2009). Model-based engineering (MBE)/model-based systems engineering (MBSE) with an integrated framework can enable the use of SBD tradespace exploration, for some situations (i.e. early-design stage with low fidelity models), in near-real time (Specking et al. 2018a). This article provides insights on using model-based design to create and assess alternatives with set-based design.

□

Contents

Introduction

SBD analyzes sets of alternatives instead of single

solutions. Sets are “two or more design points that have at least one design option in common” (Specking et al. 2018b) or “the range of options for a design factor” (Singer et al. 2017). A design factor is a “solution parameter, characteristic, or relationship that influences the design at the system level” (Singer et al. 2017). Systems engineers should develop sets determining the design factors and separating the design factors into set drivers or set modifiers. Set drivers are “fundamental design decisions that define the system characteristics that enable current and future missions,” while set modifiers are “design decisions that are ‘added on’ to the system and can be modified to adapt for new missions and scenarios” (Specking et al. 2018b).

SBD is not the best design method for every situation. SBD is particularly useful in early-stage design and if the project contains the following attributes:

- A large number of design variables,
- Tight coupling among design variables,
- Conflicting requirements,
- Flexibility in requirements allowing for trades, or
- Technologies and design problems not well understood – learning required for a solution (Singer et al. 2017)

In early-stage design, SBD helps inform requirements analysis and assess design decisions (Parnell et al. 2019). Quantitative SBD requires an integrated MBE environment to assess the effects of constraining and relaxing requirements on the feasible tradespace. For example, Figure 2 demonstrates the effects of constraining or relaxing requirements of an unmanned aerial vehicle case study with all of the explored designs in orange, the tradespace affected by non-requirement constraints (e.g. physics with requirements relaxed to not affect the tradespace) in blue, the original UAV feasible tradespace in yellow, and the relaxed (black)/constrained (red) tradespaces.

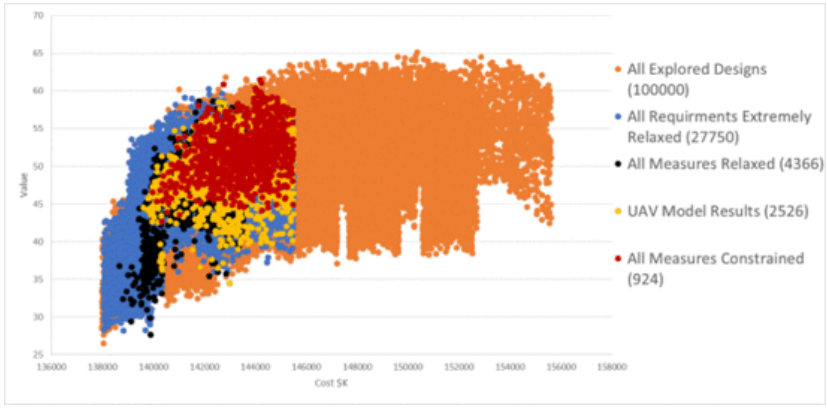

Figure 1. Effects of Requirements on the UAV's Feasible Tradespace (Parnell et al. 2019, used with permission)

The tornado diagram seen in Figure 3 shows results of a one requirement at a time analysis. This makes it easy to see how the constraining/relaxing of each individual requirement affects the feasible tradespace. Figure 3 shows that the requirements “Detect Human Activity at Night” and “Detect Human Activity in Daylight” have the greatest impact on the feasible tradespace.

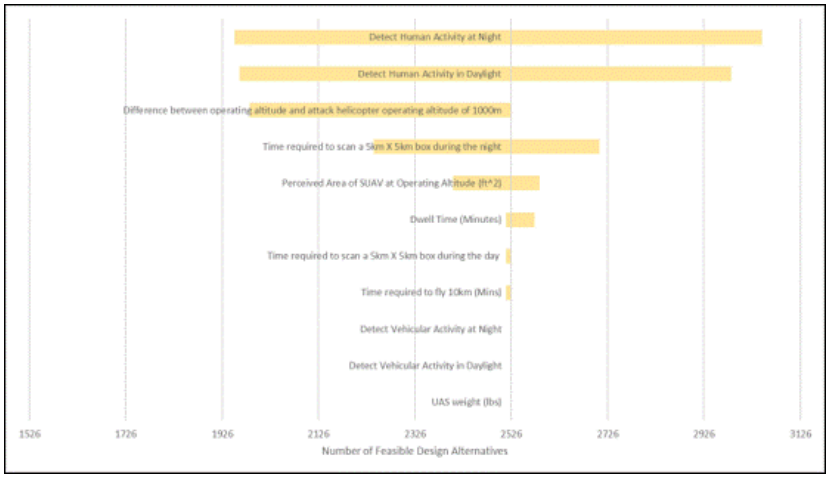

Figure 2. UAV Case Study Results of One-by-One Requirement Analysis (Parnell et al. 2019, used with permission)

Changing the requirements does not always translate to finding improved designs. The individual one requirement at a time analysis scatterplot provides important information, as seen in an example illustration in Figure 4. It is important to carefully analyze the Pareto Frontier created by each change (represented by a different color) and compare it to the Pareto Frontier of the original analysis. If the original requirement level produces better alternatives, then it does not make sense to change (constrain or relax) the requirement.

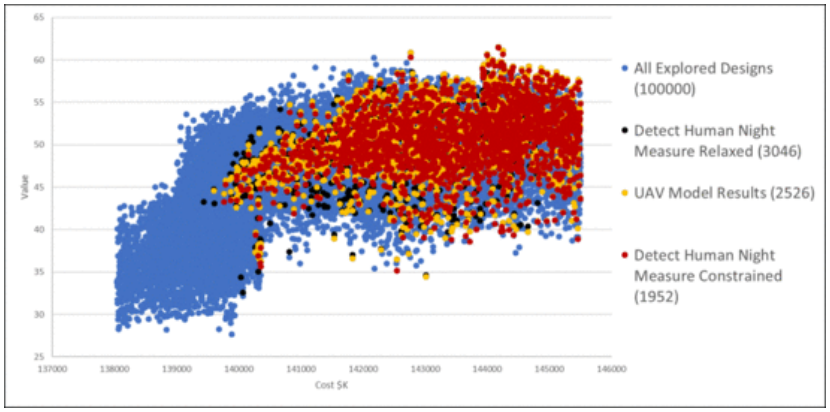

Figure 3. Effect on Feasible Tradespace by Changing Most Sensitive UAV Requirement (Specking et al. 2019, used with permission)

Additionally, using SBD can add value to the overall project and team. Some of the advantages include:

- enabling reliable, efficient communications,
- allowing much greater parallelism in the process, with much more effective use of subteams early in the process,
- allowing the most critical, early decisions to be based on data, and
- promoting institutional learning (Ward et al. 1995).

System Analyst Set-Based Design Tradespace Exploration Process

Figure 4 illustrates SBD as a concept for system design and analysis. This SBD illustration contains 5 distinct characteristics:

1. start by determining the business/mission needs and system requirements;
2. use the business/mission needs and system requirements to perform design and analysis techniques throughout time in the exploratory, concept, and development stages of the system's life cycle;
3. perform design and analysis concurrently as much as possible;
4. inform requirement analysis by using feasibility, performance, and cost data; and
5. consider a large number of alternatives through the use of sets and slowly converge to a single point solution (Specking et al. 2019).

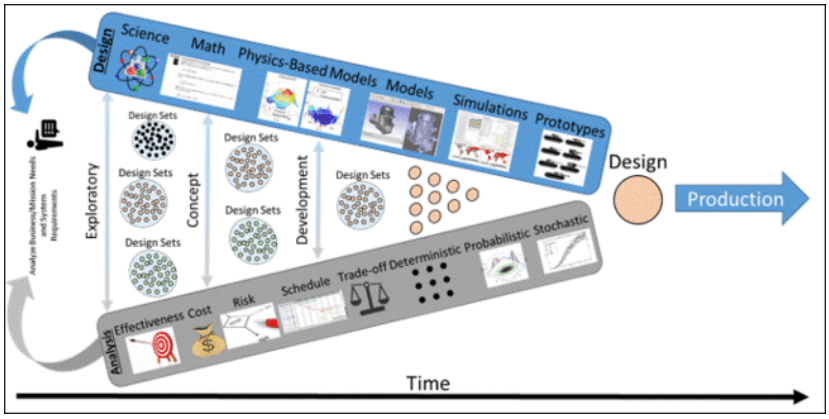

Figure 4. SBD Conceptual Framework for Systems Design
(Specking et al. 2019, used with permission)

SBD is a social-technical process and should involve input and interactions from several teams, but Figure 6 provides a SBD tradespace exploration process for system analysts (Specking et al. 2019). This eight-step process is especially useful to perform early-stage design (Specking et al. 2018b). The system analyst starts by analyzing the business/mission needs and system requirements. Systems engineers use this information, along with models and simulations developed by themselves or provided by systems and subsystem teams, to develop an integrated model. Systems engineers include requirements to assess feasible and infeasible alternatives using this integrated model. They explore the tradespace by treating each design decision as a uniform (discrete or continuous) random variable. An alternative consists of an option from every design decision. Systems engineers then use the integrated model to evaluate each alternative and to create the feasible tradespace. Monte Carlo simulation is one method that enables a timely alternative creation and evaluation process. The created tradespace will consist of infeasible and feasible alternatives based upon the requirements and any physics-based performance models and simulations. Systems engineers should work with the appropriate stakeholders to inform requirements when the tradespace produces a significantly small number of or no feasible alternatives. In addition to feasibility, systems engineers should also analyze each design decision by using descriptive statistics and other analyses and data analytics techniques. This information provides insights into how each design factor influences the feasible tradespace. Once the tradespace contains an acceptable number of alternatives, it is then classified by sets. This is an essential part of SBD. If the set drivers or design factors are not known, systems engineers should view the tradespace by each design decision for insights. Systems engineers should use dominance analysis and other

optimization methods to find optimal or near optimal alternatives based upon the measures of effectiveness. Systems engineers should explore the remaining sets for additional insights on the feasible tradespace and the requirements. The final part of this process is to select one or more sets to move to the next design-stage. It should be noted that this process contains cycles. At any part of this process, systems engineers should use the available information, such as from tradespace exploration or set evaluation, to inform requirement analysis or update the integrated model. Additionally, the systems engineer should update the integrated model with higher fidelity models and simulations as they become available. The key is to have the "right" information from the "right" people at the "right" time.

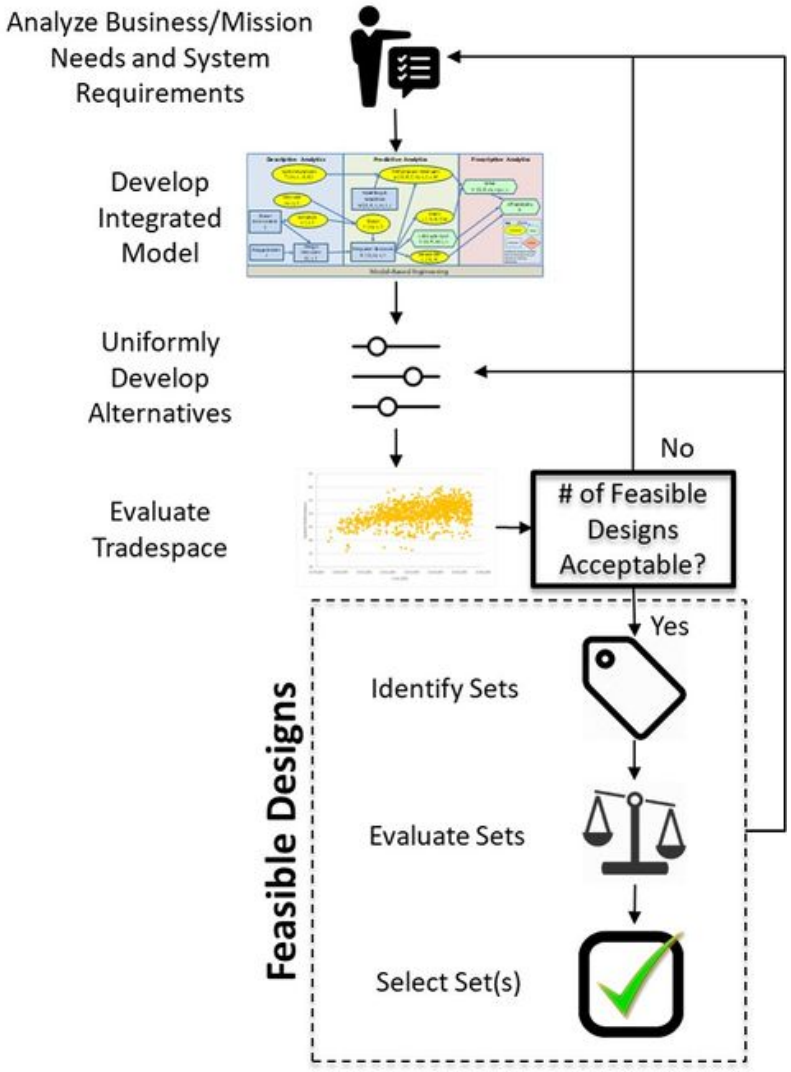

Figure 5. System Analysts SBD Tradespace Exploration Process (Specking et al. 2019, used with permission)

References

Works Cited

Parnell, G.S., E. Specking, S. Goerger, M. Cilli, and E. Pohl. 2019. “Using set-based design to inform system requirements and evaluate design decisions.” Proceedings of the 29th Annual International Council on Systems Engineering (INCOSE) International Symposium, Orlando, FL, USA, July 20-25, 2019.

Singer, D.J., N. Doerry, and M. E. Buckley. 2009. “What is set-based design?” *Naval Engineers Journal*, vol. 121, no. 4, pp. 31–43.

Singer, D., J. Strickland, N. Doerry, T. McKenney, and C. Whitcomb. 2017. “Set-based design.” *SNAME T&R Bulletin SNAME (mt) Marine Technology Technical and Research Bulletin.*

Specking, E., C. Whitcomb, G. Parnell, S. Goerger, E. Kundeti, and N. Pohl. 2018a. “Literature review: Exploring the role of set-based design in trade-off analytics,” *Naval Engineers Journal,* vol. 130, no. 2, pp. 51-62.

Specking, E., G. Parnell, E. Pohl, and R. Buchanan. 2018b. “Early design space exploration with model-based system engineering and set-based design,” IEEE *Systems*, vol. 6, no. 4, pg. 45.

Ward, A., I. Durward Sobek, J. C. John, and K. L. Jeffrey. 1995. “Toyota, concurrent engineering, and set-based design,” in *Engineered in Japan: Japanese Technology-management Practices*. Oxford, England: Oxford University Press. pp. 192–216.

Primary References

None.

Additional References

None.

< Previous Article | Parent Article | Next Article >
SEBoK v. 2.10, released 06 May 2024
