ERS Tradespace Toolset

19th Annual NDIA Systems Engineering Conference
26 October 2016

Tommer R. Ender, PhD, PMP
Principal Research Engineer
Chief, Systems Engineering Research Division
Georgia Tech Research Institute
tommer.ender@gtri.gatech.edu

Simon R. Goerger, PhD
Director, Institute for Systems Engineering Research Engineer Research & Development Center
US Army Corps of Engineers
simon.r.goerger@usace.army.mil
Summary of Topic

• Analytical building blocks
• Systems Engineering processes
• High Performance Computing
• ERS radeBuilder walkthrough
Resiliency Characteristics in an ERS Context

Serves Effectively

Trusted

In a wide range of contexts (i.e., across a variety of missions)

Easily adapted through reconfiguration and replacement

Graceful and detectable degradation of function

Through its capability and capacity to

Repel, Resist, and/or Absorb disruptions

Recover from disruptions to sustain operations and performance

Adapt to new or changed operating conditions

Exhibit Broad Utility

Effectively
Guiding Concepts

Create open SW architecture and integrated toolset

Support exploration of SE questions via higher-level DSL abstractions

Promote interoperability and extensibility

Evolve how SEs more effectively represent and express their problems

Promote collaborative design and analysis in an executable environment

Support higher-level decision making goals quantifiably and traceably
An Executable Systems Engineering Process

- Define High Level Needs
- Define, Refine and Map Requirements
- Identify or Define Analyses
- Define System
- Define and Execute Tradespace
- Verify/Validate Analyses
- Analyze Tradespace Results
- Redefine Tradespace Ranges and/or Subsystems of Interest

Requirements Loop
- Relax High Level Needs
- Verify Requirements can Distinguish between System Options of Interest
- Determine feasibility of calculating metrics for requirements, revise requirements if metrics cannot be computed

System Architected Loop
- Prioritized Requirements
- System Parameters, Ranges
- Analyses to Execute, Parameters, Ranges
- Metrics to Calculate
- Verify Requirements can Distinguish between System Options of Interest
- Determine feasibility of calculating metrics for requirements, revise requirements if metrics cannot be computed

Analysis Loop
- Tradespace Results
- Define and Execute Tradespace
- Identify or Define Analyses
- System Parameters Required for Calculations
- Define System
- Verify/Validate Analyses

Decision
Promote Collaborative Systems Engineering for the Masses

**Collaboration**
- Facilitate the integration of expertise
- Decide together

**Interactivity**
- Help understand
- Expedite the hypothesis-test-learn cycle

**Iteration**
- Lessen the effort in repeating the process
- Encourage review of knowledge

**High Performance Computing**
- Reduce time to obtain insight
- Explore the option space more fully

**Better Systems Engineering**
End-to-End Capability Supporting Diverse Stakeholders

**Stakeholders/ Users**
- Systems Engineers
- Coders/ SW Developers
- SMEs – Physics & Engineering
- Analysts / Domain, Ops Research
- Requirements Specialists
- Programmatic Specialists
- Decision Analysts

**End-to-End Capabilities**
- Requirements Definition/ Specification
- System Definition/ Specification
- Orchestration
- Data Exploration
- Decision Analysis
Deployment Use Cases

B. IN PROGRESS
- Web-based
- Remote web-server + access restricted database server
- Webserver's CPUs

C. TBD
- Web-based + HPC
- Remote web-server + access restricted database server
- Super Computers

D. IN DEVELOPMENT
- Local + HPC
- Local VM with Local Database Server
- Super Computers
Example Deployment Use Cases
Software Architecture

• **Support both regular and power users**
  - Regular users employ the more visual interface
  - Power users employ an extensible programming interface

• **Relies on permissive open source technologies**
  - Software used does not require release of own code

• **Extensible and scalable**
  - Modular architecture intended to facilitate the integration of new technologies
Execution with HPC
Pillars of Complex Decision Making

• **Define**
  users describe the needs, the analyses to assess whether or not the needs are met, and the system (or systems) being designed to satisfy said needs

• **Execute**
  users set conditions for and manage/monitor the execution of the integrated engineering models

• **Explore**
  users assess the information generated by the execution of the models to improve their mental models of the problem and the system of interest
Supporting Disciplines

• **Define**
  users describe the needs, the analyses to assess whether or not the needs are met, and the system (or systems) being designed to satisfy said needs

• **Execute**
  users set conditions for and manage/monitor the execution of the integrated engineering models

• **Explore**
  users assess the information generated by the execution of the models to improve their mental models of the problem and the system of interest
Integrate Requirements, Design, and Knowledge Generation

Collaborative SE

Scalable Framework

Dynamic, Customizable, and Visually Interactive

End-to-End Capability

Server or Stand-Alone Execution

Method and Stakeholder Fluid
TradeBuilder will automatically generate analyses based on the FAR diagrams which can be executed using the tradespace generator interface. Current functionality allows for exploring connected constraint blocks using a Latin Hypercube sampling method. Future capability will expand the tradespace generation methods to include Monte Carlo Simulations, other Design of Experiments, and optimization techniques.

Exploit the data generated by the tradespace generator using the Explore Interface. Current functionality implements a customizable dashboard with coordinated views, so as the user brushes on one plot, that selection is reflected in the others. Additionally, the tool has value transformations based on the prioritization of requirements. Future functionality will add other visualizations, e.g., Parallel Coordinates, and Scatterplot matrices.
The Problem

1. Improve Performance
   - ID: "1.1" Text: "Have hover, speed, range, payload and fuel efficiency characteristics "beyond any current rotorcraft"

2. Autonomous Operation
   - ID: "1.2" Text: "Must have optionally piloted or autonomous flight capability"

3. Hover time
   - ID: "1.1.3" Text: "Combat radius of 424 km (263 mi) with and overall unfueled range of 848 km (527 mi)"

4. Combat Radius
   - ID: "1.1.4" Text: "Carry up to 12 combat-ready troops"

5. Payload Capacity
   - ID: "1.1.5" Text: "Maximum speed of 230 km (260)"
The Analysis

par [ConstraintBlock] Performance

Main Rotor
- c_d0
- k_i
- solidity
- tip_speed
- diameter

Tail Rotor
- c_d0
- solidity
- tip_speed
- diameter

Air Vehicle
- empty_weight
- gross_weight
- tail_rotor_arm

Design Requirements
- combat_radius
- ceiling

TradeBuilder will automatically generate analyses based on the PAR diagrams which can be executed using the tradespace generator interface. Current functionality allows for exploring connected constraint blocks using a Latin Hypercube sampling method. Future capability will expand the tradespace generation methods to include Monte Carlo Simulations, other Design of Experiments, and optimization techniques.

Explore the data generated by the tradespace generator using the Explore Interface. Current functionality implements a customizable dashboard with coordinated views, so as the user changes on one axis, the selection is reflected in the others. Additionally, the tool has value transformations based on the prioritization of requirements. Future functionality will add other visualizations, e.g., Parallel Coordinates, and Scatterplot matrices.
Define your system of interest by collaboratively authoring System Block Definitions and Parametric Diagrams. Set your system KPPs and KSAAs as requirements. Navigate the project tree to view the details.

TradeBuilder will automatically generate analyses based on the PAR diagrams which can be executed using the tradespace generator Interface. Current functionality allows for exploring connected constraint blocks using a Latin Hypercube sampling method. Future capability will expand the tradespace generation methods to include Monte Carlo Simulations, other Design of Experiments, and optimization techniques.

Explore the data generated by the tradespace generator using the Explore Interface.

Current functionality implements a customizable dashboard with coordinated views, so as the user brushes on one plot, that selection is reflected in the others. Additionally, the tool has value transformations based on the prioritization of requirements. Future functionality will add other visualizations, e.g., Parallel Coordinates, and Scatterplot matrices.
Applications: Chem Bio Defense

• Application of ERS TradeBuilder to the Chem Bio defense problem

• Leveraging the OneSAF simulation framework to quantify operational effectiveness of technologies

• In coordination with the US Army Edgewood Chemical Biological Center
Applications: USAF Development Planning

• Integration with AFSIM simulation framework

• Immediate application to USAF experimentation campaign in Defeating Agile Intelligent Targets

Informs USAF planning and technology investment
Applications: Next Generation Air Dominance

• “More than just the design of a 6th generation fighter” – impact on the full Family of Systems

• Stated need to appreciate the entire life cycle up front

The United States Navy has officially kicked off its analysis of alternatives (AoA) for a future replacement for the Boeing F/A-18E/F Super Hornet strike fighter and its EA-18G Growler electronic attack derivative.

The Next Generation Air Dominance (NGAD) effort—which until recently used to be called the F/A-XX—will study a range of options to replace the service’s aging strike fighter fleet in an era of renewed great power contest where the threat environment promises to grow evermore challenging. But unlike previous Pentagon programs, the NGAD may not be a single new airframe—rather it might be a family of systems.
Acknowledgements

• Dr. Santiago Balestrini-Robinson
• Dr. Dane Freeman
• Mr. Drew Pihera
• Mr. Jason Poovey
• Dr. Valerie Sitterle

• Dr. Owen Eslinger
• Mr. Alex Baylot
• Mr. Daniel Chausse
• Mr. David Stuart
• Dr. Andrew Strelzoff

Portions of this material are based upon work supported, in whole or in part, by the United States DoD through the Systems Engineering Research Center (SERC) under Contract HQ0034-13-D-0004. SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology. The views and conclusions are those of the individual authors and participants, and should not be interpreted as necessarily representing official policies, either expressed or implied, of the DoD, any specific US Government agency, or the US Government in general.