Agenda

• EA Program (rt-125):
  — Enhancements
  — Demonstration
  — Future Plans

• Tools Development (RT-123):
  — Experience Design
  — Simulation
  — Learning Assessment
Experience Accelerator (RT-125)
Experience Accelerator Program Overview

• **What:** SERC Multi-year Phased Research Program

• **Sponsor:** Defense Acquisition University

• **Collaboration:** Stevens, Georgia Tech, Purdue, USC (year 1)

• **Problem Statement:** *Traditional Systems Engineering (SE) education is not adequate to meet the emerging challenges posed by ever increasing Systems and Societal demands, the workforce called upon to meet them and the timeframe in which these challenges need to be addressed.*

• **Program Goal:** *Transform the education of SE by creating a new paradigm capable of accelerating the time to mature a senior SE while providing the skills necessary to address emerging system’s challenges.*
Hypothesis: By using technology we can create a simulation that will put the learner in an experiential, emotional state and effectively compress time and greatly accelerate the learning of a systems engineer faster than would occur naturally on the job.

Goals: To build insights and “wisdom” and hone decision making skills by:

- Creating a “safe”, but realistic environment for decision making where decisions have programmatic and technical consequences
- Exposing the participants to job-relevant scenarios and problems
- Providing rapid feedback by accelerating time and experiencing the downstream consequences of the decisions made
The Experience: A Day in the Life of a PSE

UAV System:
- S0 – System
- S1 – Airframe and Propulsion
- S2 – Command and Control
- S3 – Ground Support

UAV KPMs:
- Schedule
- Quality
- Range
- Cost

Phases:
- EA Introduction
  - Phase 0: New Employee Orientation
- Experience Introduction
  - Phase 1: New Assignment Orientation
- Experience Body
  - Phase 2: Pre-integration system development -> CDR
  - Phase 3: Integration -> FRR
  - Phase 4: System Field Test -> PRR
  - Phase 5: Limited Production and Deployment
  - Phase 6: Experience End
- Experience Conclusion
  - Phase 7: Reflection

Each session = 1 day
BLOOMS TAXONOMY

EVALUATION
- Assessing theories; Comparison of ideas;
- Evaluating outcomes; Solving; Judging;
- Recommending; Rating

SYNTHESIS
- Using old concepts to create new ideas;
- Design and Invention; Composing; Imagining;
- Inferring; Modifying; Predicting; Combining

ANALYSIS
- Identifying and analyzing patterns;
- Organisation of ideas; recognizing trends

APPLICATION
- Using and applying knowledge;
- Using problem solving methods;
- Manipulating; Designing; Experimenting

COMPREHENSION
- Understanding; Translating;
- Summarising; Demonstrating; Discussing

KNOWLEDGE
- Recall of information;
- Discovery; Observation;
- Listing; Locating; Naming
Targeted Lessons

• Problem solving and recovery
  — Identify weight and drag problems, remediate with TPM targets and allocation changes
  — Identify schedule problem, remediate with additional staff and a small schedule delay
  — Identify software quality problem, remediate with increased software design review frequency

• Product integration
  — All sub-systems need to be done at the same time ideally for integration to begin. Adding resources to airframe/propulsion (AP) and command/control (CC) sub-systems brings their schedule more in line with the other sub-system and the systems integrator, who are not having schedule issues.
  — The solution could be improved by not hiring as many AP senior staff and reducing staff for the systems integrator and ground station sub-contractor so that they meet their targets at the three-month delay, instead of the original CDR schedule. This is left as an exercise for the learner and/or instructor.
  — Transferring weight allocation from AP to CC illustrates the relationships between these two sub-systems and how a win-win can be achieved (or at least a win and no-loss).

• Cutting corners to make short term goals while ignoring long term outcomes
  — Make decisions early, even though they have negative cost implications initially. This is better than facing the bigger problems later on of schedule delays that make the cost overruns worse.
Learning Results

Results demonstrate that for two targeted learning outcomes in particular, nearly all of the teams learned the outcomes as they very clearly highlighted them in their presentations as the lessons they had learned. Other learning objectives were also highlighted by the different teams.

- These lessons were learned despite the fact that the learners only fully completed the first two phases of the experience before speeding through the remaining phases in order to see their results.

- Furthermore, the EA was designed to be played multiple times by learners, so these results are indicative of impressive learning gains given the limited implementation of the experience.
Pilot System Evaluation

The instructor familiarization, instructor and student pilots:

• provided feedback affirming the potential of the EA with students in a classroom,

• generated 140+ action items for EA enhancements,

• proved the need for stability and performance improvements.

These items are being addressed in RT125 Increment 3.
EA Enhancements

• A stable multi-player capability. This capability shall allow experienced instructors to provide feedback to the student teams, such as explaining the outcomes of the teams’ decisions, and helping the students understand the impact of their recommendations.

• The capability to provide dialog based on the students’ past recommendations.

• Improvements to the user interface to allow students to quickly find and analyze information through status charts and dashboard.

• Refined simulation mode, enabling trades between range, loiter time and payload weight.

• Performance and stability enhancements.

• A predictive capability to the SEEA to improve the students’ decision-making capabilities and produce better recommendations.

• Some additional discrete, technical/program tradeoff decisions.
Multi-Player

Modes:
• Single Learner mode
• Single Learner with supervisor (PM & Mentor)
• Multiple Learner
• Multiple Learner with supervisor

Prototype multi-learner capabilities have been developed for the Experience Accelerator. The capabilities have been provided for multiple learners to create and join games asynchronously, share documents, communicate directly with one another, and make decisions that affect the outcome of the simulations.
Experience Accelerator
Welcome Jon Wade
Your Experience Awaits You!

Option Menu
(please select choice below):
- UAV Experience
- Multiplayer
- Profile Update
- Logout

UAV Experience Status
- New User Orientation: Complete
- New Experience Orientation: Complete
- Pre-Integration System Development: Active
- System Integration
- System Field Test
- Limited Production and Development
- Experience End
- Reflection

Help
Abort Experience
Future Plans
Program Goals

1. Integrate EA into SYS30x

• EA System Capabilities
  — Completion and stabilization of multi-learner mode
  — Provide means of informing learner of impact of recommendations
  — Ensure that dialog is synchronized with recommendations
  — Improve learner interface with status charts to eliminate need to page through entire set

• Deployment Deliverables
  — Define explicit EA deliverables to support DAU deployment

• Hosting Requirements
  — Specify technical details of hosting requirements
2. Create additional EA experiences deployed at multiple sites
   — Develop joint competency to fill out EA experience through the life-cycle
   — Develop 2-3 other experiences for different domains
   — Trial tools through these experience development efforts
   — Integrate EA into SERC Collaborator courses
   — Validate through usage with INCOSE workgroups

3. Create sustaining open source development community
   — Develop tool suite using SERC core funding
   — Release EA technology, tools and experiences to open-source community
   — Create consortia to provide long-term support
Funding & Joint Research

• Extended Capabilities
  — DAU: Option Year 3, Pilot + Multi-Learner Technology Support
  — SERC: Content Creation Tools funding

• New Experiences
  — DAU: Logistics Experience, proposal submitted
  — ONR: Team experience, white paper submitted
  — NSF: Learning in Formal and Informal Settings, proposal submitted
  — NRO: Spacecraft experience, intend to pilot SEEA
  — MITRE: Team experience, discussions
  — LMC: Interest
  — SI Corporation: Proposal interest
  — Sponsored doctoral research: 2 Stevens students
EA User Group Meeting - Attendees

- Al Stanbury    ARDEC
- Chris Robinson DAU
- John Snoderly DAU
- Aileen Sedmak DoD DRE
- Mimi Heisey    LMC
- Marilee Wheaton NRO/Aerospace
- Regina Griego  Sandia
- Bill Miller    Stevens
- Doug Bodner    Georgia Tech
- Bill Watson    Purdue
- Rich Turner    Stevens
- Jon Wade       Stevens

First meeting held on 11/25/201. Strong interest from many member to evaluate EA in their organizations to determine future deployment.

**Action Items:**
- Set up EA User Group using a website for communications.
- Set up EA workshop at INCOSE IS.
- Create Evaluation Terms of Agreement.
- Schedule next EA User Group meeting.
Experience Accelerator Tools (RT-123)
# Motivation for Tools

<table>
<thead>
<tr>
<th>#</th>
<th>Limitation</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Complex simulation models with limited reuse supported</td>
<td><strong>Sim Builder</strong> - Simulation model builder utilizing libraries/templates</td>
</tr>
<tr>
<td>2</td>
<td>Complex simulation outputs dependent on hundreds of variables/parameters</td>
<td><strong>Sim Tuner</strong> - Parameter tuner that automates the tuning of parameters to yield desired outputs via batch processing of different combinations</td>
</tr>
<tr>
<td>3</td>
<td>Manual nature of phase and cycle development</td>
<td><strong>Phase Editor</strong> - GUI-based tool for phase, cycle and event specification with code generation</td>
</tr>
<tr>
<td>4</td>
<td>Manual nature of specifying events and their triggers in the Experience</td>
<td><strong>Event Editor</strong> - GUI or text-based tool to specify events and their triggers with code generation</td>
</tr>
<tr>
<td>5</td>
<td>Manual nature of artifact integration involving re-linking and recompilation</td>
<td><strong>Artifact Integrator</strong> - Artifact entry application that allows designer to take an artifact file and enter it into EA application with automatic recompilation and re-linking</td>
</tr>
<tr>
<td>6</td>
<td>Manual nature of assessment of learner performance</td>
<td><strong>Learning Assessor</strong> - Assessment tool-suite that provides automated performance scoring, decision comparisons against proven baselines, etc.</td>
</tr>
</tbody>
</table>
Tools in Context of EA Architecture

Experience Accelerator Block Diagram

- **Learning Assessor**
- **Artifact Integrator**
- **Phase Editor & Event Editor**
- **Sim Builder & Sim Tuner**
Phase Editor

- Automate creation of experience phases
- Microsoft Visio-type interface
- Set state (e.g., phase starting time, variable values)
- Phase ordering (repeat, go back, skip, etc.)
- Available events per phase
- Available NPCs per phase
- Available documents per phase
Event Editor and Artifact Integrator

• Event Editor
  — A tool that provides the capability to create and edit events during an experience and the activities that may trigger them.
  — For example, a phone call from the learner’s supervisor can be triggered based on a decision made by the learner or the state of the project.

• Artifact Integrator
  — A tool that allows an experience builder the ability to quickly upload an experience change.
  — For example, a new artifact can be uploaded, such as a new dialog or report, or a changed phase and or event.
  — The developer can then test the results without having to do any programming.
Basic Simulator Functions

• Represents the state over time of the system being developed
  — Technical performance measures and key performance parameters
  — Quality

• Represents the state over time of programmatic elements
  — Cost (EVM)
  — Schedules (milestones such as entrance criteria for review meetings)
  — Staffing

• Advances these states via behavioral models and relationships
  — Feedback loops
  — Lags
  — Non-linear behavior

• Implemented using system dynamics formalism
Decision Support

• Provides graphical output to learner for decision support

• Learner decisions fed into simulator to change trajectory of behavior
  — Add resources
  — Focus on technical problems via diverting resources
  — Change allocations between sub-systems

• Cycle-based learning and problem-solving

![Simulator](image1)

<table>
<thead>
<tr>
<th>Overall System</th>
<th>Learner Decisions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule:</td>
<td></td>
</tr>
<tr>
<td>Confidence Level to Achieve Program Schedule Goals</td>
<td>&lt;H,M,L&gt;</td>
</tr>
<tr>
<td>Actions to address issues:</td>
<td></td>
</tr>
<tr>
<td>Nothing Required</td>
<td>○</td>
</tr>
<tr>
<td>Call in external audit team</td>
<td>○</td>
</tr>
<tr>
<td>Add senior/junior design staff</td>
<td>Sr○/Jr○</td>
</tr>
<tr>
<td>Add development equipment</td>
<td>○</td>
</tr>
<tr>
<td>Add facilities</td>
<td>○</td>
</tr>
<tr>
<td>Reduce capabilities</td>
<td>○</td>
</tr>
<tr>
<td>Anticipate schedule extension by xx months</td>
<td>&lt;xx&gt;</td>
</tr>
</tbody>
</table>

![Status & Trends](image2)
• Simulation models present substantial development and curation challenges
  — Expert domain knowledge and simulation knowledge required
  — Validation against authentic and compelling experience targets versus development time and effort
  — These issues have been known and largely unsolved for many years

• Simulation models have complex parameter and variable relations, making desired output behavior difficult
  — Hundreds of variables and parameters
  — Complex inter-relations
  — Non-intuitive behavior
  — Learner inputs provide numerous additional “degrees of freedom” beyond the baseline simulation that need to be understood
Overall Success Metrics

• Creation and tuning of models with correct intended behavior/outputs

• Intuitive support for model building constructs
  — Dynamics
  — Dependency
  — Feedback
  — Lags

• Usability consistent with general software usability norms and guidelines

• Sub-model archiving capability and limited library of sub-models

• Support for model integration with EA application
• System dynamics models typically use XML schema
• Trend is toward standard schema with specializations for different vendors
Simulation Model Using GUI
Use Cases – Conceptual Design

• Design model structure
  — Specify sub-models and sectors, plus relationships between them
  — Specify learner decisions and mapping to variables (including constraints on decisions and relations between different ones)

• Evolve an existing model
  — Add new elements to an existing model
  — Modify existing elements
  — Change challenge level / add landmines

• Design models consistent with an existing model, but corresponding to future/past phase
  — Common variables and behavior
  — Time-based differences
Other Use Cases

• Sub-Models
  — Create a sub-model
  — Design sub-model interfaces
  — Validate a sub-model
  — Archive a sub-model
  — Tune an existing sub-model
    o Internal parameters
    o External relations
  — Add an archived sub-model to a new model

• Usability
  — Create graphical model
    o Place elements
    o Connect elements
    o Manipulate elements
  — Specify math equations
  — Standard functionality
    o Save, edit, cut-copy-paste, zoom

• Outputs
  — Design charts
    o Variables, plans, units, labels, time horizon
  — Test consistency of actual not-intervention output vs. desired
  — Test consistency of actual learner decision output vs. desired
Usability Heuristic Priorities

• User control and freedom
  — Making sure user can exit unintended state without having to go through an extended dialog

• Consistency and standards
  — Platform-specific standards for design

• Error prevention

* Nielsen
Sim Builder Design

• Features adapted from commercial packages

• Concept of Models Vs. Maps
  — Instead of trying to explain each node on the model itself, explanation done on separate copy of the model “map”

• How sub-models are handled within the overall model
  — Breaking sub-models up into sectors.
  — Linking sectors using one node in one sector from another sector
  — Documentation functionality for sub-models (description, key parameters, behavior, linkages)
  — GUI and XML support required

• Other features (from EA)
  — Tag learner-controlled variables (recommendation form), along with allowable values
  — Design landmines into simulations
  — Specify challenge levels
Sub-Model Example
Sim Tuner

- Excel-type interface
- Read in variables and chart specs from XML
- One tab for the Graphs and one for Variables
- Values are entered on the Variables tab
- Switching from Variables to Graphs tab causes graphs to recalculate themselves if new values were entered.
- Also, incorporate cancel button that brings you back to main view
Learning Assessor

• Develop tools that provide the means to measure learning through performance, actions and self-assessments:
  — Measure quantitative score of simulation results
  — Compare decisions and actions of the students with those of experts
  — Review students’ evaluation of the lessons learned
  — Compare changes in the above results through students’ iterations in the Experience
  — Perform longitudinal studies to assess how learning is transferred to the workplace
Thoughts?
Join the Experience Accelerator Team!

Contact for information:

Jon Wade & Rich Turner
jwade@stevens.edu
rturner@stevens.edu

Doug Bodner
doug.bodner@gatech.edu

Bill Watson
brwatson@purdue.edu

This material is based upon work supported, in whole or in part, by the Defense Acquisition University through the Systems Engineering Research Center (SERC). SERC is a federally funded University Affiliated Research Center (UARC) managed by Stevens Institute of Technology in partnership with University of Southern California.
Backup Slides
How easy is it to create new experiences?
Leverage Points

- Uses current virtual desktop and other login/logout etc. technology
- Uses existing Single or Multi-User mode
- Project is composed of subsystems with similar simulation requirements to the existing prototype (overall system, software centric C&C mechanical/electrical system, human based ground support)
- Critical parameters of schedule, cost, quality are used
- Lifecycle based approach start after the completion of the CONOPS, requirements and project plans
- Utilizes the same or similar project phasing
- Uses current learning cycle process: learner recommendation, feedback and evaluation process
- Learning is taking an acquisition view on the program
- Similar charts are provided by the simulation for program status
- Uses current challenges and lessons learned
- Uses similar dialog
- Uses similar recommendation and other forms
Experience Conceptual Design

• Objectives
  — Determine the specific competencies, ‘aha’s’ and targeted difficulty levels to be supported
  — Define the program’s high-level performance measures, their targets that constitute success and deviations from target that cause failures

• Story line
  — Select/design a particular scenario (e.g., a particular DoD program) for the experience that supports the desired learning and develop a storyboard for the desired experience elements
  — Define phases and cycles within phases

• Story line (cont.)
  — Define challenges and landmines consistent with competencies, ‘aha’s’ and difficulty levels
  — Specify important non-player characters/roles, plus any live player roles

• Technology Support
  — Specify desired simulated program behavior, status outputs, inter-relations between elements, state variables, and learner control points
  — Specify desired artifacts (e.g., program background material, learner decision/recommendation forms, etc.)
  — Specify the types of feedback to be provided to the learner based on decisions made during the experience
<table>
<thead>
<tr>
<th>System</th>
<th>Challenge</th>
<th>Phase</th>
<th>Evidence</th>
<th>Situation</th>
<th>Desired Actions</th>
<th>Inputs to Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>S2</td>
<td>range too short</td>
<td>P2</td>
<td>range projections</td>
<td>weight during development is too high</td>
<td>Reallocate resources - focus resources on weight reduction</td>
<td>Change subsystem allocation - reallocate weight from S2 to S1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Change weights</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Change range target</td>
</tr>
<tr>
<td>S1</td>
<td>range too short</td>
<td>P3</td>
<td>range projections</td>
<td>drag is higher than expected in wind tunnel testing</td>
<td>Reallocate resources - focus resources on drag reduction</td>
<td>Change assignment of labor in S1</td>
</tr>
<tr>
<td>S1, S2</td>
<td>schedule</td>
<td>P2</td>
<td>completion rates</td>
<td>productivity lower than expected</td>
<td>Add resources - hire additional labor</td>
<td>Hire new personnel</td>
</tr>
<tr>
<td>S2</td>
<td>schedule</td>
<td>P3</td>
<td>completion rates</td>
<td>more changes had to be made than anticipated</td>
<td>Adjust schedule</td>
<td>Change schedule target</td>
</tr>
<tr>
<td>S0</td>
<td>schedule</td>
<td>P3</td>
<td>completion rates</td>
<td>unexpected integration issues</td>
<td>Add resources - hire additional labor; purchase additional test equipment</td>
<td>Hire new personnel for S0; add test equipment resources</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reallocate resources - focus on integration, get help from other areas</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Change assignment of labor in S0</td>
</tr>
<tr>
<td>S2</td>
<td>quality</td>
<td>P2, P3</td>
<td>defect rates</td>
<td>software defect rate is too high</td>
<td>Reallocate resources - focus resources on design/code reviews</td>
<td>Change labor assignment</td>
</tr>
</tbody>
</table>
Experience Development and Testing

• Develop
  — Develop/write desired artifacts (e.g., program background material, learner decision/recommendation forms, etc.)
  — Develop artifacts to be populated with simulation output to provide learner insight into current and previous program status
  — Develop non-player characters and state-based dialog whereby the learner can query the NPCs to discover additional information
  — Implement phase and cycle behavior
  — Develop simulation models and datasets to ensure desired simulated program behavior and inter-relations of program elements (via testing/tuning)

• Develop (cont.)
  — Embed challenges and landmines into simulation models and NPC dialog
  — Write scripted feedback to learner based on alternate learner decisions, linked to program outcomes
  — Integrate artifacts, simulation models, NPC dialogs, and learner feedback into Experience Accelerator

• Test consistency of experience
  — Iteratively test with SMEs in development process
  — Test with instructors
  — Evaluate with students in pilots
  — Enhance Experience based on feedback
GUI Dialog Tool – Chat Mapper <improved slid
Dialog Statistics

- Capabilities
- Dependencies/decisions
- Number of dialogs
- Number of nodes
## EA Tools

<table>
<thead>
<tr>
<th>#</th>
<th>Limitation</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Complex simulation models with limited reuse supported</td>
<td><strong>Sim Builder</strong> - Simulation model builder utilizing libraries/templates</td>
</tr>
<tr>
<td>2</td>
<td>Complex simulation outputs dependent on hundreds of variables/parameters</td>
<td><strong>Sim Tuner</strong> - Parameter tuner that automates the tuning of parameters to yield desired outputs via batch processing of different combinations</td>
</tr>
<tr>
<td>3</td>
<td>Manual nature of phase and cycle development</td>
<td><strong>Phase Editor</strong> - GUI-based tool for phase, cycle and event specification with code generation</td>
</tr>
<tr>
<td>4</td>
<td>Manual nature of specifying events and their triggers in the Experience</td>
<td><strong>Event Editor</strong> - GUI or text-based tool to specify events and their triggers with code generation</td>
</tr>
<tr>
<td>5</td>
<td>Manual nature of artifact integration involving re-linking and recompilation</td>
<td><strong>Artifact Integrator</strong> - Artifact entry application that allows designer to take an artifact file and enter it into EA application with automatic recompilation and re-linking</td>
</tr>
<tr>
<td>6</td>
<td>Manual nature of assessment of learner performance</td>
<td><strong>Learning Assessor</strong> - Assessment tool-suite that provides automated performance scoring, decision comparisons against proven baselines, etc.</td>
</tr>
</tbody>
</table>
Simulation
Learner Interaction

<table>
<thead>
<tr>
<th>Schedule:</th>
<th>Overall System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confidence Level to Achieve Program Schedule Goals</td>
<td>&lt;H,M,L&gt;</td>
</tr>
<tr>
<td>Actions to address issues:</td>
<td></td>
</tr>
<tr>
<td>Nothing Required</td>
<td>○</td>
</tr>
<tr>
<td>Call in external audit team</td>
<td>○</td>
</tr>
<tr>
<td>Add senior/junior design staff</td>
<td>Sr/Jr</td>
</tr>
<tr>
<td>Add development equipment</td>
<td>○</td>
</tr>
<tr>
<td>Add facilities</td>
<td>○</td>
</tr>
<tr>
<td>Reduce capabilities</td>
<td>○</td>
</tr>
<tr>
<td>Anticipate schedule extension by xx months</td>
<td>&lt;xx&gt;</td>
</tr>
</tbody>
</table>

Learner Recommendations

Systems Dynamics Simulations
- Java engine
- XML model files
- XML chart specification files

Current Status (State Variables, Program Impact Reports & State-based NPC Dialog)
System Dynamics Simulation

- Stock-and-flow representation
- Continuous variables with discrete extensions
- Feedback loops representing complex interdependencies
  - Positive feedback for unboundedness
  - Negative feedback for equilibrium
- Lags, Brooks Law and other organizational phenomena
- Applications
  - Software development processes
  - Earned value management
  - Market and macro-economic behavior
  - Epidemics and other population phenomena
Modeling Advances

- Physics-based model of aerodynamic KPPs/TPMs
- Sub-system manufacturing and assembly into full systems
- State-based resource commitments
  - Allocation of staff to reduce weight growth if weight becomes a problem
- Statistics (PDR to CDR):
  - 171 input/output variables
  - 100 intermediate state variables
  - 156 state change equations
- Upcoming:
  - Discrete design trades
Pre-Integration Phase Model
Model Development

System dynamics model GUI

XML code

XML editor
## Generic Simulation Sub-Models

<table>
<thead>
<tr>
<th>Sub-Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work breakdown structure (WBS)</td>
<td>Sub-system models predominate in design and pre-integration phases, while the system model becomes more predominant in pre-production development.</td>
</tr>
<tr>
<td>Technical performance</td>
<td>Technical performance of the system and sub-systems is modeled using a physics-based representation that links KPPs and underlying TPMs (with values going from estimates to actuals).</td>
</tr>
<tr>
<td>Entrance criteria</td>
<td>Progress is modeled for on meeting various entrance criteria (e.g., digital design drawings complete) for review meetings (e.g., critical design review). Each review has a number of entrance criteria.</td>
</tr>
<tr>
<td>Quality</td>
<td>Defect generation, identification and resolution.</td>
</tr>
<tr>
<td>EVM</td>
<td>The EVM sub-model accrues cost over time according to categories of labor, equipment and capital deployed in the program. A management reserve is supported.</td>
</tr>
<tr>
<td>Workforce</td>
<td>Each sub-system is developed by a different systems engineering workforce (i.e., sub-contractor).</td>
</tr>
</tbody>
</table>
# UAS Experience Sub-Models

<table>
<thead>
<tr>
<th>Sub-Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work breakdown structure (WBS)</td>
<td>Work breakdown structure (WBS) for different sub-models: Airframe &amp; propulsion (S1), command &amp; control (S2), ground station (S3).</td>
</tr>
<tr>
<td>Phases</td>
<td>Phases for different development stages: Pre-integration development (up to critical design review), post-integration development (up to flight readiness reviews), flight test (up to production readiness review), low-rate initial production.</td>
</tr>
<tr>
<td>Technical performance</td>
<td>Technical performance metrics: KPP = range; TPMs = drag, propulsion efficiency, weight (broken down by sub-system).</td>
</tr>
<tr>
<td>Entrance criteria</td>
<td>Entrance criteria: Six to seven criteria per phase as a subset of standard criteria used for the review meeting.</td>
</tr>
<tr>
<td>Quality</td>
<td>Quality metrics: Command &amp; control software defects and resolution.</td>
</tr>
<tr>
<td>EVM</td>
<td>EVM metrics: Standard EVM metrics used for each sub-system and overall program (actual cost of work performed - ACWP, budgeted cost of work performed - BCWP, budgeted cost of work scheduled - BCWS).</td>
</tr>
<tr>
<td>Workforce</td>
<td>Workforce: Workforce is modeled for each sub-system, with a workforce for system integration. Workforce can be allocated to various functions in each sub-system.</td>
</tr>
</tbody>
</table>
Sub-Model Relationships

- Range is a function of weight, drag and propulsion efficiency
  - Weight is contributed by different sub-systems, each of which has weight growth

- Progress on entrance criteria is affected by the design and development workforce levels and mix of relevant sub-system sub-contractors, as well as resource assignment (e.g., weight or drag reduction).

- Progress on identification and remediation of software defects in the command & control sub-system depends on the quality assurance workforce level and design review frequency.

- Either junior or senior engineering workforce levels can be changed by setting a target staff level. There are lags between the initiation of hiring and improved progress due to additional resources.

- Hiring additional workforce means that labor costs increase, typically causing ACWP to increase relative to BCWP (cost variance), but potentially narrowing the gap between BCWS and BCWP (schedule variance).

- Instead of hiring, the schedule can be extended, delaying review meetings. And resulting in additional accrual in ACWP during the schedule slippage.
Airframe & Propulsion TPMs

- Moderate weight growth
  - Above plan based on historical weight growth
  - Below allocation

- Drag and propulsion efficiency degrading

- Setting targets for drag and propulsion efficiency
  - Diverts senior staff from other tasks (e.g., entrance criteria for CDR)
  - Improves TPM performance

- Reallocating weight to command & control sub-system
  - Diverts senior staff to control weight due to reduced margin
  - Reduces weight growth rate
Command & Control TPMs

- Unusually large weight growth
  - Well above plan based on historical weight growth
  - Exceeds original allocation
  - Increased software complexity requires more processors, power units, cooling systems, etc. This was unanticipated in original plan.

- Reallocating weight to command & control sub-system
  - Allows fewer senior staff needed to control weight
  - Allows weight growth to be accommodated
Weight growth is causing the range KPP to fall short of the design objective and even the requirement.

Weight re-allocation and targets for drag and propulsion efficiency can salvage the situation, if done early enough.
Errors are generated, identified, then resolved.

There are two categories of errors – critical and non-critical.

Unresolved critical errors are well above the historical error target.

Increasing the frequency of design reviews causes the error generation rate to be reduced due to more rigorous design, thus leading to fewer unresolved errors.

Alternatively, staff could be increased to address errors.
• Progress toward completing entrance criteria is modeled.
• Comparison to the plan indicates whether development is ahead or behind schedule.
• If behind schedule, one option is to increase staff.
  — Brooks Law is modeled, so productivity drops as more people are added.
  — Junior staff require training and mentoring (senior staff time). They are also not as productive as senior staff.
  — There is a limited pool of talent that the program can bring on board.
  — There is a lag between the decision to staff up and the start of new staff.
• Both airframe & propulsion and command & control have higher than normal levels of junior staff at the outset (50% and 35%, respectively).
Each sub-contractor has its own EVM modeled, as does the prime.

These roll up into an overall program EVM.

BCWS, BCWP and ACWP are tracked.

The prime has a program-level management reserve.

Staffing decisions influence BCWP and ACWP.

Schedule variance, cost variance, SPI and CPI are tracked, as well.

These are used as performance indicators on the program dashboard.
Simulation Tools

- **Sim Builder** - Simulation model builder using libraries/templates

- **Sim Tuner** - Parameter tuner that automates the tuning of parameters to yield desired outputs via batch processing of different combinations of settings

- **Goal** - Provide experience developers with tools to facilitate developing simulation for different experiences
Dialog
Non-Player Character (NPC) Engine

• Each NPC is an object with independent state and behavior

• NPC behavior is largely composed of Player/NPC dialogue, plus reaction to game state
  — Represents dialogue as finite-state machine (game developers call “hub-and-spoke”)
  — Reacts to game state by “registering interest” in state variables through publish & subscribe

• NPC Engine is a generalized graph-walker, holds state machines as dynamic data structures
Hub-and-Spoke Dialogue

- **Hub** is a conversation state
  - Point within a larger dialogue context
  - Associated with chunk of dialogue spoken by NPC

- **Spokes** are conversational alternatives
  - Options that move the dialogue into a new state
  - Selected by Player during conversation
  - Associated with chunk of dialogue spoken by Player

- **Conversation** is a path through hubs along spokes
How a Conversation Unfolds

- As each dialogue hub is entered...
  - NPC “speaks” an element of dialogue
  - Engine presents dialogue alternatives to Player
  - Player chooses among alternatives, “speaks” associated dialogue
  - Engine enters new dialogue hub, and cycle repeats
Dialogues in DAU Prototype

• The PM, ex-Gov LSE, current Prime LSE are all available for some portions of Phase 1 and Phase 2

• Currently over 500 dialogue nodes (hub and spoke) have been defined

• Currently all dashboard displays and recommendations are available for use in determining conversation path

• Currently only the phase and cycle number are available as state variables
Example of Chat Mapper© Dialogue Structure for a Prototype Character
Emails are Conversations, Too!

• Dialogue model extensible to other communication modes, such as email

• Open “dialogue channel” to NPC through email
  — NPC Engine enumerates dialogue alternatives for current state of target NPC
  — Instead of “speaking” dialogue, Player assembles alternatives into outgoing email
  — NPC responds by assembling reply from path through dialogue graph, dictated by alternatives Player has chosen